History of Far Eastern Languages in Computing

カタカナ

漢字

Looking to East Asia
by Kurt Hensch
2004

IBM History of Far Eastern Languages in Computing

National Language Support Since 1961

Kurt Hensch

Notice:
This is the second version of this book, after pre-publication of a private edition dated 2003-May-28. This second edition consolidates miscellaneous updates and expands its contents in concert with a three-part series of papers submitted to the IEEE Annals of Computing.

Document Information:
As this book is written in English combined primarily with Japanese, the ideal tools for producing it were IBM ThinkPads, Japanese version. Running under the Japanese version of Windows 98 and later XP Professional, data entry, editing, and processing of bi-lingual text was done using the Japanese versions of Microsoft Word and Excel. The entire book is one MS Word file, all figures inserted. A PDF file was created using the Adobe Acrobat software, to facilitate printing on various devices.

For photographs and some tables that could not be made by MS Excel, an IBM Flat Bed Scanner was used, in combination with Adobe Photoshop software to prepare inserts. Some old text from archive documents was converted into MS Word files via PAGIS optical character recognition.

ISBN 3-937267-03-4

Editor (Private Edition): Kurt Hensch
e-mail: kurt@kdd.net

Picture Sources: All pictures individually released by the indicated sources

Printing, Binding and Distribution: Roehm TYPOfactory Marketing GmbH,
Boeblinger Straße 68
D-71060 Sindelfingen, Germany
e-mail: gerhard.mezger@roehmszbz.de
Printed in Germany, 2004

The letters of the top line in the yellow field on the book front cover are the Japanese characters カタカナ, pronounced "katakana", Japanese phonetic characters. The two large characters 漢字 underneath are pronounced "kanji", and they are the term used for the ideographs in Japanese and Chinese. This artwork symbolizes the range of steps in the development of computer features to handle Far Eastern Languages – from a modest start with phonetics only, to full-fledged Far Eastern Kanji language capabilities.

滝

This character is one of the few original Japanese Kanji that did not originate from the Chinese set of Hanzi. Its meaning in English is "Waterfall", pronounced "Taki", and is my wife's name. Her love was what inspired me to study the Japanese language, culture and history.

I dedicate this book to Taki, in gratitude for her care, patience, support, encouragement, and sacrifices in sharing her life with me.

Contents

Introduction

Today, millions of people in China, Japan and Korea keep the Internet humming. Websites are built and surfed; e-mail messages and files, many with photographs attached, are transmitted back and forth. Messages on command buttons and pull-down menu items, icon titles, error messages and prompts are written in their native languages. Only commands in text interface are still written in English in most cases. The people of these countries combined represent a very large percentage of the total world population. Most of them are of the young generation, with students and school children leading the group. They take it for granted that the system they are using came about automatically, along with the invention of the computer. Far from it!

The purpose of this book is to provide an overview of the efforts and events on the part of IBM that led to today's versatility in handling East Asian languages in the information technology area. This book will trace back the history over more than forty years, by chronologically highlighting major events and breakthroughs that occurred on the road to achieving the present capabilities, in three major parts.

Part 1 (pages 7 through 34) provides the analysis to the extent necessary of the Japanese and other Far Eastern languages versus existing technologies of the early 1960s, leading to the emergence of phonetic-based initial IBM products in the ensuing decade. Supported by top management of the IBM Corporation, the Product Development Laboratory of IBM Germany started the work in 1961 by developing a Katakana feature for the IBM 1401 Data Processing System.

Part 2 (pages 35 through 60) describes IBM's successful efforts during the 1970s to develop the first commercial general Kanji Computer System, and computerized Newspaper Publishing Systems in Japan.

Part 3 (pages 61 through 100) documents IBM Japan's taking the lead in the coordination of world-wide efforts in IBM, which were launched during the 1970s, in order to ensure implementation of Far Eastern language requirements with IBM products, and to eventually move into a leading role in this area of the emerging Information Technology industry. IBM Japan is providing significant resources for the development of support for Asian language capabilities in subsequent IBM products.

Prologue

Early October 1961 on a military airfield at Cold Bay Alaska, my plane landed to refuel on the way from New York to Tokyo. This was my first trip to East Asia ever. Jet planes were still rare at that time; there was no non-stop connection. I flew the last leg of my journey on a Canadian Pacific Airlines Britannia turboprop aircraft that I had boarded in Vancouver BC at the Pacific coast.

With World War II a fresh memory, I felt very uneasy being on the most western point of the United States mainland where the Aleutian chain of islands extends towards Japan. Some of them actually were under Japanese occupation during the war. Not knowing what to expect in the country I was about to visit, I did not realize that I was at the most significant turning point of my career, and in my whole life. At Haneda Airport, Toby Takahashi from IBM Japan Commercial Relations, and Jack Manning, his counterpart at IBM Asia Pacific Headquarters (APHQ), met me. I felt relieved, because without their escort, I would not have made it to my hotel. I became determined to study the language, in order to be better prepared for subsequent trips.

On an assignment at IBM World Trade Corporation Headquarters in New York for half a year, I was given the special task of acting as liaison engineer to the Oki Electric Company in Japan. IBM had entered a contract with them for the development of a computer printer. The machine was considered to become a lower cost/speed alternative to the IBM 1403 chain printer. Eventually, a prototype was shipped to the IBM Product Development Laboratory in Endicott NY for evaluation. However, the printer did not perform to specifications, and the contract with the Oki Electric Company was terminated.

During my visits to Japan in pursuit of the Oki Electric printer liaison project, I met several employees of the IBM Japan Sales organization. While the IBM1401 System had soon become a bestseller in the western world, they expressed their frustration to me about the difficulty in marketing IBM products that can function only by using the English language. The implementation of national language requirements was mandatory if the domestic market potential of non-English speaking countries was to be fully exploited. A proposed Katakana feature for the IBM 1401 System was the first step, though ever so modest.

I learned that Koichi Tanaka, a senior manager of that group, was trying to coordinate the development of a Katakana feature for the IBM 1401 system, in collaboration with the Product Development Laboratory of IBM Germany. The reason was that IBM Japan would not have its own Product Development Lab for many more years to come. Tanaka's contact in the German Lab was Immanuel Witt, and two engineers, Fritz Rausch and Werner Hasler. They developed the necessary hardware modifications.

Through my contacts with many individuals from IBM Japan, I became very interested in Japan as a country, its culture, history and language. This created a strong motivation in me to assist in the coordination effort for the IBM 1401 Katakana project between IBM Japan and the Product Development Lab of IBM Germany. As a result, Dr Karl Ganzhorn, its director, assigned me to the task of "Project Manager Katakana" in 1961, to lead and coordinate all required development efforts.

Figure 1. IBM Germany Development Laboratory 1963 Source: Ref. 20

Managing this project required worldwide coordination. IBM Japan Sales documented the product objectives for the IBM 1401 Katakana system and other necessary peripheral equipment, such as a Katakana keypunch and verifier. The German Lab developed the character set code structure, the hardware modifications to the Central Processing Unit (CPU), and software support. As the IBM 1403 was developed there, the IBM Endicott (New York) Product Development Lab designed the Katakana type slugs for the IBM 1403 Chain Printer and its electronic controls. The Toronto IBM Manufacturing Plant was responsible for production. IBM announced the 1401 Katakana feature in the fourth quarter 1963.

The entire year 1962 was indeed a travel year for me, flying around the globe several times. IBM announced the 1401 Katakana feature in the fourth quarter 1963. **Figure 1** shows Fritz Rausch and me in the Product Development Laboratory of IBM Germany, examining an IBM 1403 Chain Printer Katakana test printout.

This project revealed the obvious need for an engineering planning function in the IBM World Trade Asia area for national language requirements not only in Japan, but also for other markets in Asia. In fall 1963, I was named Development Engineering Advisor to IBM APHQ in Tokyo, reporting to the Director of Development Engineering at IBM World Trade Headquarters in New York.

Characters Used in Far Eastern Countries

This book focuses on the written languages in Japan, Korea, China, and Thailand. These are countries with significant Information Technology market potential, and use languages that are much different and considerably more complicated than those of other countries. This overview will briefly explain the basic structures of these four languages, while subsequent sections will mainly focus on Japan.

Japan Script

Unlike all languages with numerals and alphabets of a relatively small, fixed number of characters, the Japanese language has three sets. All, plus Roman letters, are used in written Japanese.

テレビジャパンは **NHK** の番組を中心にライブニュース、ドラマ、スポーツ、ドキュメンタリー、邦画、子供番組など、ご家族全員がお楽しみ頂ける番組を一日中お届けします。
テレビジャパンは **NHK** の番組を中心にライブニュース、ドラマ、スポーツ、ドキュメンタリー、邦画、子供番組など、ご家族全員がお楽しみ頂ける番組を一日中お届けします。
TV JAPAN offers full day programming daily. From latest news, drama, sports, documentary, to children's programs, you will find it on TV JAPAN.

Figure 2. Sample Japanese Sentence Source: Ref. 21

The example in the top box of Figure 2 illustrates this complexity. The same sentence is shown in the center box, but with Katakana printed red, Hiragana blue, Kanji black and Roman green. NHK stands for 日本放送協会 (Nippon Hoosoo Kyokai), which means Japan Broadcasting Corporation. The bottom box provides the meaning of this sentence in English.

Hiragana and Katakana are phonetic sets with 87 characters each. Katakana is largely used for foreign words and for printing on low cost devices that have no Kanji. Hiragana is primarily used to create grammatical variations to words containing Kanji characters.

Several thousand Kanji, pictorial ideographs of Chinese origin, exist of an indefinable number that reaches into five digits. There are several

Kanji subsets, defined for various levels in schools, for newspapers, official government use, etc. Nobody would be able to name the absolute total number of Kanji characters.

The existence of phonetic sets facilitates the representation of words by pronunciation only. However, ambiguities frequently occur, because there are many different Kanji ideographs with the same pronunciation, but with different meanings. Further, most Kanji characters in Japan have at least two pronunciations (on-yomi and kun-yomi), some even more. Combinations of two or more Kanji may form another meaning, but emerge with a combined pronunciation that may not resemble any of those of the single Kanji used. Examples of resulting ambiguities are shown in Figure 3. The people of Japan deserve high admiration for being able to learn and master their language.

Kanji		**Pronunciation**		
Character(s)	**Meaning**	**On**	**Kun**	**Other**
東	east	too	higashi	azuma
南	south	nan	minami	
京	capital	kyoo	miyako	
貴	noble	ki	tooto(i)	
雄	hero	yuu	o, osu	
貴雄	noble hero	kiyuu		takao
上	top, above	joo	ue	a, kami
大	big, great	dai, tai	oo	
和	peace, harmony	wa	yawa(ragu)	kazu
大和	great harmony	daiwa		yamato

Note: Parentheses items are Hiragana extensions

Figure 3. Kanji Pronunciation Ambiguities

The existence of phonetic sets facilitates the representation of words by pronunciation only. However, ambiguities frequently occur, because there are many different Kanji ideographs with the same pronunciation, but with different meanings. Further, most Kanji characters in Japan have at least two pronunciations (on-yomi and kun-yomi), some even more. Combinations of two or more Kanji may form another meaning, but emerge with a combined pronunciation that may not resemble any of those of the single Kanji used. Examples of resulting ambiguities are shown in Figure 3. The people of Japan deserve high admiration for being able to learn and master their language.

The phonetic scripts in Japan are called Hiragana and Katakana,. Figure 4 shows the entire sets with a color-coded indication of those letters that were made available on the IBM 1401 system. Certain groups of letters were omitted and had to be substituted by similar other letters. It was a compromise in order to keep the character set small and, therefore, to maintain an acceptable printing speed on the IBM 1403 printer.

The complete Katakana set consists of 81 characters, those of Sets 1, 2, and 4 together. These are too many relative to both coding in a 6-bit byte structure, as well as substituting for the 26 upper case Roman alphabet letters. It was determined that, by omitting some characters that are either rarely used, or that could be substituted by others but still retain the meaning of words, a reduced Katakana set of 47 letters could be implemented (Sets 1 and 3).

Letters of the omitted categories were ヲ(wo), and the entire Set 4, which are smaller size versions of the letters ア イ ウ エ オ ヤ ユ ヨ ワツ (a i u e o ya yu yo wa tsu). These letters are needed to represent the pronunciation of certain Kanji characters. On the IBM 1401 system, all these had to be printed using full size letters, because encoding and type slugs for the correct, small letter versions (Set 4) could not be provided.

An analog example of a sentence in English with such compromise could perhaps look like this: "The incRease of pRinT speed is achieved by subsTiTuTing The loweR case leTTeRs T and R wiTh TheiR capiTal veRsions". This example pretends that the letters R and T are available only in upper case, while the rest of the alphabet is lower.

Also not implemented on the IBM 1401 system were all letters of Set 2, being 25 consonants the pronunciation of which can be modified by accents such as ガ(ga) ギ(gi)グ(gu) ゲ(ge) ゴ(go) パ(pa) ピ(pi) プ(pu) ペ(pe) ポ(po). Instead, it was deemed acceptable to print the basic consonant, and the accents on the subsequent position (example: ホ°instead of ポ, or コ" instead of ゴ). This required the two accents (Group 3) to occupy a separate type slug each, although they are not part of the official Katakana character set.

The Hiragana set shown in the five right columns of Figure 4 can be looked at as round-shaped phonetic symbols with a one-to-one correspondence to the printed Katakana set and, therefore, were not

considered for computer implementation at that time. The number of letters and their pronunciation are identical.

Set	Pronunciation					Katakana / Hiragana									
1	a	i	u	e	o	ア	あ	イ	い	ウ	う	エ	え	オ	お
	ka	ki	ku	ke	ko	カ	か	キ	き	ク	く	ケ	け	コ	こ
	sa	si	su	se	so	サ	さ	シ	し	ス	す	セ	せ	ソ	そ
	ta	ti	tu	te	to	タ	た	チ	ち	ツ	つ	テ	て	ト	と
	na	ni	nu	ne	no	ナ	な	ニ	に	ヌ	ぬ	ネ	ね	ノ	の
	ha	hi	hu	he	ho	ハ	は	ヒ	ひ	フ	ふ	ヘ	へ	ホ	ほ
	ma	mi	mu	me	mo	マ	ま	ミ	み	ム	む	メ	め	モ	も
	ya		yu		yo	ヤ	や			ユ	ゆ			ヨ	よ
	ra	ri	ru	re	ro	ラ	ら	リ	り	ル	る	レ	れ	ロ	ろ
	wa				wo*	ワ	わ							ヲ*	を
	n					ン	ん								
2	ga	gi	gu	ge	go	ガ	が	ギ	ぎ	グ	ぐ	ゲ	げ	ゴ	ご
	za	zi	zu	ze	zo	ザ	ざ	ジ	じ	ズ	ず	ゼ	ぜ	ゾ	ぞ
	da	di	du	de	do	ダ	だ	ヂ	ぢ	ヅ	づ	デ	で	ド	ど
	ba	bi	bu	be	bo	バ	ば	ビ	び	ブ	ぶ	ベ	べ	ボ	ぼ
	pa	pi	pu	pe	po	パ	ぱ	ピ	ぴ	プ	ぷ	ペ	ぺ	ポ	ぽ
3						゛		゜							
4						ァ	ぁ	ィ	ぃ	ゥ	ぅ	ェ	ぇ	ォ	ぉ
						ャ	ゃ			ュ	ゅ			ョ	ょ
						ヮ	ゎ			ッ	っ				

Set	Description
1	45 Basic full size Katakana, * letter "w o" omitted
2	25 Full size Katakana, pronunciation modified by accents
3	2 Katakana accents as a separate letter to modify pronunciation
4	10 small size letters to modify pronunciation of consonants

Figure 4. Katakana and Hiragana Character Sets

The exclusive use of phonetic symbols, however, would not be sufficient. During my first visit to Japan in 1961, I found out about a society that promotes the introduction of the sole use of Katakana for written Japanese. I saw one of their periodicals called “Kana no Hikari” (Ray of Katakana). All articles there were written in Katakana only.

Most Japanese people were very skeptical, though, because of the ambiguities of meaning versus pronunciation as explained earlier. It would have been too much wishful thinking to expect that the use of Kanji would soon disappear. On the contrary, forty years later with

Kanji capability even in the smallest PC, it seems that Kanji will most likely remain in use forever.

The Korean Hangul may be somewhat of a comparable case, although slanted much stronger towards the use of Jamo. However, the Korean national language character set still includes a large number of Kanji (or Hanja).

China Script

Mao Ze Dong changed many Kanji ideographs (pronounced "Hanzi" by natives) on Mainland China, while Taiwan kept the "traditional" characters. Only the shape of the character was changed, or "simplified", while the meaning remained the same. Two distinctly different sets emerged, "Simplified Chinese" (SC) and "Traditional Chinese" (TC).

The number of Chinese Hanzi is even larger than that of Kanji in Japan. Some analyses indicate that the number can be over 80,000. There is one advantage with Chinese compared to Japanese, though, as each Hanzi has basically only one pronunciation.

請在此頁下面查找你的投票站地址
Check Your Polling Place Address Below

English Meaning	Traditional Chinese		Simplified Chinese		Japanese with Pronunciations		
	Hanzi	Pronounce	Hanzi	Pronounce	Kanji	On yomi	Kun yomi
broad, wide	廣	guang	广	guang	広	kou	hiroi
air	氣	qi/ci	气	qi	気	ki	
child	兒	er	儿	er	児	ji	ko
country	國	guo	国	guo	国	koku	kuni
put in motion	發	fa	发	fa	発	hatsu	
close	關	guan	关	guan	閉	hei	tojiru

Figure 5. Chinese Sample Sentence and Kanji Modifications

The upper box of **Figure 5** shows a sentence written in Traditional Chinese with its English meaning underneath, taken from a voter

instructions flyer for the Chinese community in San Francisco. The lower box illustrates how original Hanzi shapes changed while being simplified, or adopted as Japanese Kanji, by retaining the meaning. These are just a few examples that do not even scratch the surface of the complexity.

Korea Script

Figure 6 provides some Korean examples with their meaning in English underneath. Both parts of Korea apply this concept of using both, Hangul (left side of the slash) or Hanja (right side of the slash).

The Korean character set is yet again different from Japanese and Chinese. Korea has 3 scripts. One is Kanji, which the natives call "Hanja". The second is "Jamo", which is a set of consonants and vowels. There are 93 Jamo characters, a few of which are to be positioned into designated parts of a single character print/display area in both horizontal and vertical directions to compose a single character.

Six specific rules prescribe two to four Jamo that can compose a character that is then called Hangul (the third script). They are not ideographs, as a Hangul character represents only pronunciation and not a meaning. Under these rules, 2620 Hangul characters are being used, although the theoretical total permutations would exceed 10,000.

The Names of Two Countries
대한민국/大韓民國 Republic of Korea, ROK (South Korea)
조선 민주 주의 인민 공화국/朝鮮民主主義人民共和國 Democratic People's Republic of Korea, DPRK (North Korea)

Figure 6. Korean Samples Source: Ref. 2

Thailand Script

Although the shape of Thai character symbols is completely different, there are conceptual similarities to the Korean Hangul. Components from different groups are placed into one print or display position under certain rules to construct a Thai symbol. There are 87 components, which includes consonants, vowels, numbers, and some special characters. However, the number of valid composed Thai symbols amounts to over 2000 permutations. See Figure 7 for a sample Thai phrase, with English in the adjacent box.

The concept of written Thai text is to place the components on their designated invisible horizontal lines, as illustrated in Figure 8. The base line (1) is used where consonants and middle vowels are placed. The line below the base line (2) is used for writing lower vowels. The line above the base line (3) is used for writing upper vowels. The line above the upper vowel line (4) is used for tone marks. If there is no upper vowel, the tone mark is written on the upper vowels line (3).

ข้อมูลเกี่ยวกับภาษาไทย	Information about the Thai language

Figure 7. Sample Thai Phrase Source: Ref. 17

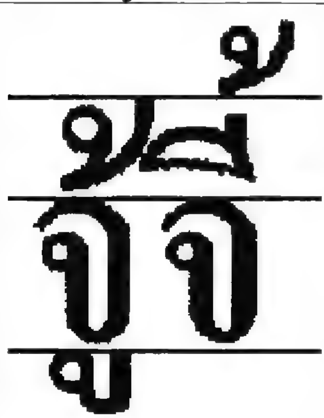
Composed Thai symbol	Positioning of Thai components to form a composed symbol
	4. Tone mark symbol position 3. Upper vowel symbol, diacritics or tone mark symbol position 1. Base line symbol and Western alphabet position 2. Lower vowel position

Figure 8. Structure of a Thai Composed Symbol Source: Ref. 17

Burma, Laos, Cambodia

The Burmese, Lao, and Khmer (Cambodia) languages are related to Thai. Among these, Lao is most similar to Thai. Burmese and Khmer share their language root with Bali and Sanskrit, similar to Thai. The only characteristic, which their writing systems share with Thai, is that they require kerning (shape-changing) as well.

Technologies Required for Handling Characters

Fixed versus Complex Character Sets

From their inception through the 1950's, the development of computers and the accompanying software has almost entirely been based on the English language and several special characters. The technology at that point facilitated the design of storage, display, printing, and data entry devices to accommodate the 10 numerals, the 26 letters of the English alphabet and some special characters.

Other industrialized countries accepted this situation, but quickly developed their own national versions to a certain extent, and without much additional effort. Thus, computer systems "speaking" German, French, Spanish, etc. were up and running. Even several non-Latin character sets like Russian and Greek belonged to this category, because they all have limited, explicitly defined character sets.

The situation was quite different in the major East Asian countries, and data processing users had been living with computing systems in an "English mode" only, for decades. Suitable technologies simply did not exist that would facilitate the design of equipment that could fully satisfy their national language requirements. The Chinese, Korean, Japanese, and Thailand markets required the following concepts and technologies:

Coded character set structures to identify thousands of characters.

- Memories large enough to accommodate huge font libraries and dictionaries.

- Processors fast enough for quick font retrieval and running of certain necessary programming routines to, for example, reconcile ambiguities, and issue prompts for character selections.

- Keyboard designs for the data entry of thousands of different characters.

- All points addressable (APA) high-resolution displays.

- High-speed printers for thousands of different characters with reasonable speed.

Actual Technology Status

IBM had announced the IBM 1401 System in 1959 with its unique IBM 1403, 600 lines per minute chain printer. Using transistor circuits Standard Modular System (SMS) technology for the first time instead of vacuum tubes, the system was compact and energy efficient. It was praised as a powerful system, having "a very large memory of 4K"! Just compare such flimsy storage space with today's minimum requirement of about 500K 8-bit bytes to house 7000 Kanji in a 24x24 dot matrix.

Yet, the IBM 1401 system became a bestseller and success worldwide. Programming languages in fashion were Assembler, COBOL (Common Business Oriented Language), and RPG (Report Program Generator), all of course English-based. Major non-English countries quickly moved and substituted their own national language character sets to satisfy local customers, except Japan, China, Korea and Thailand.

The 1401 6-bit binary coded decimal (BCD) code was sufficient for identifying the upper case of the limited number of characters in the English alphabet, plus numbers and some special characters. Card codes (hole patterns) for all 64 BCD positions had to be maintained. The IBM 026 Card Punch had a feature to punch card codes for Katakana BCD positions.

It must be kept in mind that in the mid sixties, punched cards were widely used as the main data storage media. About twenty years later, however, they completely disappeared, because low cost storage technologies for large volumes of data became available. Figure 9 shows a blank 80-column IBM punch card. Figure 10 shows a card with rectangular holes punched with the card codes that represent the data interpreted at the upper edge of the card. The cards are rotated by 90 degrees in order to show them in approximately 90% of actual size.

Punch cards came in shoebox size containers, with 2000 cards in one such box. With 80 columns in a card, one card could store 80 card codes or characters. This means that 2000x80 = 160,000 characters, or 160K bytes occupied the space of a shoebox! It further means that about nine shoeboxes full of punch cards were needed to store what fits in a (now old fashioned) floppy disk of 1.4 megabytes capacity.

IBM developed and maintained a large line of products for punching, interpreting, reading, sorting, collating, and merging punched cards. Today, they could be found only as museum pieces. Likewise, samples of punch cards were difficult to find. Fortunately, Toshiaki Igi could locate

some by searching amongst his IBM colleagues and their archived documentation.

With the IBM 1401 announcement, the Marketing organization of IBM Japan demanded a national language version. Ending the continued neglect of their native language and imposition of English was essential. Realizing that the present technology was insufficient to create a full-fledged Japanese IBM 1401 System, they requested to at least do something by using the phonetic Katakana character set. In 1961, the operations of IBM Japan were limited to the Manufacturing, Marketing, and Service organizations. IBM product development resources located in another country had to be involved. Thus, the Product Development Laboratory of IBM Germany started the task of developing a Katakana feature for the IBM 1401 System.

In 1959, the status of coded Japanese character sets in electronic engineering was highly proliferated. Some companies began to implement original coded Japanese character sets for their own computers, from around 1960. However, those were incompatible amongst each other and to IBM's.

Establishment of the first Japanese Industrial Standard (JIS) for a national coded character set was a decade away. Therefore, IBM had to find its own solution for encoding Japanese characters, initially based on the IBM 1401 system architecture.

Figure 9. Blank IBM Punch Card

Source: Ref. 13

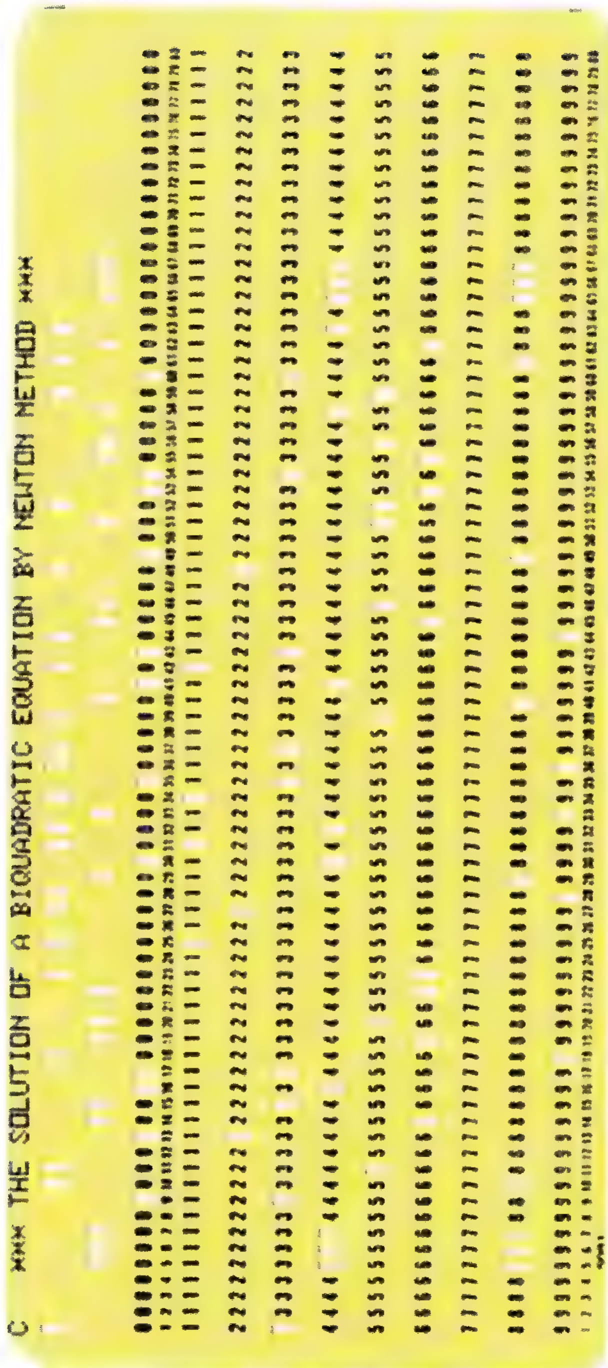


Figure 10. IBM Punch Card with Data Source: Ref. 13

Reduced Phonetic Character Set Implementation、 1960s

IBM 1401 Katakana Feature, 1963

The Binary Coded Decimal (BCD) 6-bit concept imposed a limit of using the minimum subset of 47 characters. The IBM 1401 memory was modified to accommodate the 47 Katakana characters in addition to the English alphabet, by making use of the word mark core memory plane. Written Japanese does not require spaces between words, nor hyphenation. Alternate addressing of either character set was accomplished by means of a shift-in code for Katakana, and a shift-out code. The 47 shaded fields in Figure 11 are the Katakana assignments.

This IBM effort of defining a coded character set was not the first to implement Katakana in Japanese information processing. Some other companies had already developed their own coded character sets when IBM started its Katakana implementation.

Data entry in the IBM 1401 generation had to be accomplished by means of punched cards. A modification on RPQ basis ("Request for Price Quotation"; this was a simplified method in IBM to expedite the development of special products under a shorter time frame and procedure) to the IBM 024 Card Punch was provided that could punch the Katakana character BCD card codes. Katakana character engravings were provided on the keytops, following a customary arrangement for Katakana typewriter keyboard layouts. Punching of the combination of holes for the three shift codes was added.

The basic IBM 1403 print chain consisted of 240 type slugs with four sets of 60 letters each. Interchangeable English-Numeric and Katakana-Numeric chain cartridges were provided for the IBM 1403 printer, resulting in a 600 lines per minute print speed. However, the speed slowed down if type slugs of both sets had to be provided on the same chain for text in Katakana-English mixed mode. Theoretically, 240 different type slugs could be on the chain. But, the print speed would then drop to 150 lines per minute, while 240 slugs could still not accommodate all Kanji, which are in the thousands.

Due to the rotating chain and mechanical type slug hammers on the IBM 1403, the resulting noise was considerable. It posed a constant headache to its designers to stay within acceptable maximum limits. Millions of dollars were spent for building acoustic test chambers at the IBM development labs.

Together, this concept was a barely acceptable, minimum approach for the Japanese market. Engineers of the IBM Endicott NY Development Laboratory (where the IBM 1403 was developed) jokingly called it "cut-a-corner", a sound very similar to when saying "Katakana" with slurry American accent.

IBM 1440 Katakana System, 1964

Soon after the IBM 1401 System, the IBM 1440 was announced, adding the IBM 1443 Bar Printer option in addition, or instead of the IBM 1403 Chain Printer. Character sets, print speeds, and software support were similar as in the IBM 1401 Katakana version.

Masumi Iwao's assignment during 1963-1964 to the US included rendering assistance in the product development of the Katakana feature for the IBM 1443 Bar Printer (the output device of the IBM 1440 Katakana feature).

Special efforts were necessary to develop table lookup routines for the translation from card codes to EBCDIC and vice versa.

IBM System/360 Katakana, 1964

The announcement of the IBM System/360 introduced the 8-bit Extended Binary Coded Decimal Interchange Code (EBCDIC). Figure 13 is a reproduction of the actual code sheet that I had prepared for the Japanese requirements in 1965. Apart from one quadrant reserved for controls, three with 64 code positions each became available for graphics, as opposed to the only one of the 6-bit BCD mode shown in Figure 11.

In EBCDIC, the Katakana set was assigned to codes of a separate quadrant instead of being overlaid on the English set. Further, with more code positions available in the 8-bit structure, the 10 small Katakana graphics could be assigned their own codes (Set 4 of Figure 4), together with some unique Japanese punctuation marks.

Enough space was available to use unique code positions for the 25 Katakana with accents (Set 2 of Figure 4). However, IBM Japan decided to continue to compose them by means of separately printed accents (EBCDIC positions 190 and 191). Some more code positions were available for selected Kanji, used in billing, banking, and address printing applications.

1-4 Bits 5,6->	00	01	10	11
0000	0 space	16 & ソ	32 -	48 0
0001	1 A ア	17 J タ	33 /	49 1
0010	2 B イ	18 K チ	34 S ヘ	50 2
0011	3 C ウ	19 L ツ	35 T ホ	51 3
0100	4 D エ	20 M テ	36 U マ	52 4
0101	5 E オ	21 N ト	37 V ミ	53 5
0110	6 F カ	22 O ナ	38 W ム	54 6
0111	7 G キ	23 P ニ	39 X メ	55 7
1000	8 H ク	24 Q ヌ	40 Y モ	56 8
1001	9 I ケ	25 R ネ	41 Z ヤ	57 9
1010	10 ¢ コ	26 ! ノ	42 ユ	58 : レ
1011	11 .	27 $ ¥	43 ,	59 # ロ
1100	12 < サ	28 *	44 % ヨ	60 @ ワ
1101	13 (シ	29) ハ	45 — ラ	61 ' ン
1110	14 + ス	30 ; ヒ	46 > リ	62 = ゛
1111	15 | セ	31 ¬ フ	47 ? ル	63 " ゜

Figure 11. Katakana 6-Bit BCD Code

Source: Ref. 20

Thus, two different Katakana code schemes emerged, called “Old Katakana” for the IBM 1401, and “New Katakana” for the IBM System/360. This incompatibility created difficulties in data portability, but the need for shift codes was eliminated.

Decentralized direct data entry via keyboards attached to cathode ray tube (CRT) monitors was not widely available through the late seventies. Also, memory space in central processing units was of premium value and scarce. Consequently, data entry still relied heavily on the punched card media in the IBM System/360 era. Therefore, the combinations of punched holes in IBM punched cards for each of the 256 EBCDIC codes had to be defined. Figure 14 shows those of the largest Japanese set.

Since the Katakana and Kanji characters were assigned to vacant positions scattered across the 256 fields, the derived card hole combinations were not contiguous and appear highly illogical. As punched cards were the main media for data entry, an RPQ modification to the IBM 029 Card Punch was provided, which could punch all Katakana card codes.

The new Katakana and Kanji code assignments presented several unique problems for hardware and software engineers. Besides the IBM Card Punches, other card handling equipment had to be upgraded to accommodate many new card codes that came along with the 256 EBCDIC code structure.

As many as six holes had to be punched in a single column for several characters. Stronger magnets were needed in IBM Card Punches, and those machines emitted excessive noise. Mechanisms in card reading equipment had to be modified to recognize the assigned codes correctly.

Katakana Use in Programming Languages

The first attempt to introduce Katakana in programming languages was made in 1969 at the IBM Development Lab at Hursley, UK, which had the responsibility for the PL/I Programming Language. By placing substitute Katakana words in the English keywords compiler table, Katakana keyword source statements could be translated into machine language. Some sample applications were written in “Kana PL/I” and tested. Toru Takeshita documented the results of an experimental programming course in Katakana PL/I, which were subsequently published in 1969 (Ref. 26).

Universal Character Set Printing

One way to optimize the print speed despite mounting sets with more than 60 different characters on the print chain was the "Preferred Character Set" concept (Figure 12). John Underwood, an engineer at the Product Development Lab of IBM Endicott NY, invented this method. In a circular symmetrical arrangement, he placed more slugs of characters on the chain that statistically have a high frequency of usage, and fewer slugs of less frequently used ones.

John Underwood also designed the electronic control circuits called "Universal Character Set", in which the codes of the letters arranged on the chain are reflected. Thus, it became possible to define and control a large number of different preferred character set configurations. This gimmick even allowed including a small number of specific Kanji characters for printing addresses, bills and bank statements. Such a small set of 19 selected Kanji was made available in one of the "Preferred Character Set" solutions shown in Figure 15.

John Underwood also designed the electronic control circuits called "Universal Character Set", in which the codes of the letters arranged on the chain are reflected. Thus, it became possible to define and control a large number of different preferred character set configurations. This gimmick even allowed including a small number of specific Kanji characters for printing addresses, bills and bank statements. Such a small set of 19 selected Kanji was made available in one of the "Preferred Character Set" solutions shown in Figure 15.

Preferred		Characters			Lines per Minute
Set	Frequency	Graphics	Count	Total	
A	6	10 Numeric, 2 Specials .,	12	72	692
B	2	45 Katakana, 2 Accents, 4 Special - ¥ / * 15 Kanji 都県部市区町村港()社店所株入	66	132	272
C	1	26 Alpha, 6 Specials +%@#&| 4 Kanji 計年月日	36	36	143
		Total	114	240	

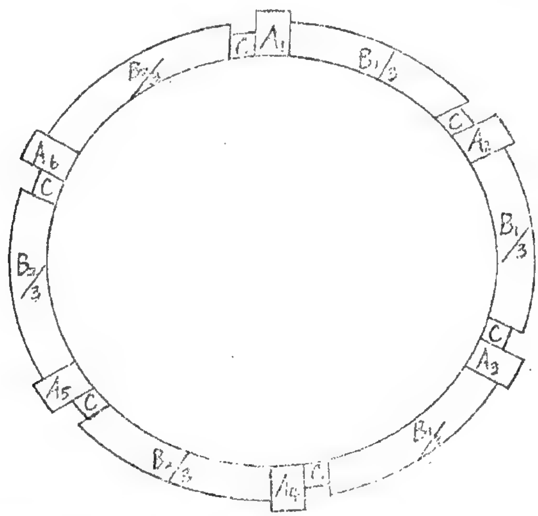


Figure 12. Preferred Character Set Concept Source: Ref. 20

The assigned codes shown in Figure 14 present the maximum of Katakana and Kanji characters, which could be made available for the Japanese market. Customers could choose from a variety of Preferred Character Set configurations. A set for the IBM 1403 printer could either be limited to the minimum Katakana (as for the IBM 1401 system) or include the System/360 expanded version, plus none, some, or all of the EBCDIC assigned Kanji.

A faster "Train Printer" with also 240 type slugs, loose in a guide instead of mounted on a chain, supplemented the IBM 1403 Chain Printer.

	00				01				10				11			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	16	32	48	64 SP	80 &	96 -	112 号	128 年	144 ソ	160 所	176 都	192 東	208 有	224 住	240 ø
0001	1	17	33	49	65 。	81 ェ	97 /	113 局	129 ア	145 タ	161 社	177 株	193 A	209 J	225 生	241 1
0010	2	18	34	50	66 「	82 ォ	98 地	114 中	130 イ	146 チ	162 ヘ	178 府	194 B	210 K	226 S	242 2
0011	3	19	35	51	67 」	83 ャ	99 式	115 小	131 ウ	147 ツ	163 ホ	179 県	195 C	211 L	227 T	243 3
0100	4	20	36	52	68 、	84 ュ	100 会	116 円	132 エ	148 テ	164 マ	180 市	196 D	212 M	228 U	244 4
0101	5	21	37	53	69 ・	85 ョ	101 支	117 入	133 オ	149 ト	165 ミ	181 郡	197 E	213 N	229 V	245 5
0110	6	22	38	54	70 ヲ	86 ッ	102 大	118 部	134 カ	150 ナ	166 ム	182 区	198 F	214 O	230 W	246 6
0111	7	23	39	55	71 ァ	87 道	103 殿	119 課	135 キ	151 ニ	167 メ	183 町	199 G	215 P	231 X	247 7
1000	8	24	40	56	72 ィ	88 字	104 出	120 丁	136 ク	152 ヌ	168 モ	184 村	200 H	216 Q	232 Y	248 8
1001	9	25	41	57	73 ゥ	89 残	105 商	121 目	137 ケ	153 ネ	169 ヤ	185 番	201 I	217 R	233 Z	249 9
1010	10	26	42	58	74 ¢	90 !	106 方	122 :	138 コ	154 ノ	170 ユ	186 レ	202 京	218 協	234 産	250 資
1011	11	27	43	59	75 .	91 $¥	107 ,	123 #	139 月	155 日	171 店	187 ロ	203 済	219 注	235 昭	251 財
1100	12	28	44	60	76 <	92 *	108 %	124 @	140 サ	156 計	172 ヨ	188 ワ	204 未	220 上	236 和	252 品
1101	13	29	45	61	77 (	93)	109 _	125 '	141 シ	157 ハ	173 ラ	189 ン	205 錠	221 売	237 名	253 分
1110	14	30	46	62	78 +	94 ;	110 >	126 =	142 ス	158 ヒ	174 リ	190 ゛	206 貸	222 買	238 阪	254 型
1111	15	31	47	63	79 |	95 ¬	111 ?	127 "	143 セ	159 フ	175 ル	191 ゜	207 借	223 下	239 行	255 用

Specials and Small KATAKANA — Basic KATAKANA — KANJI

Figure 13. IBM System/360 EBCDIC Table

Source: Ref. 20

EBCDIC Number	Katakana Characters Number	Graphic	Punched Card Code
65	1	。	12-0-9-1
66	2	「	12-0-9-2
67	3	」	12-0-9-3
68	4	、	12-0-9-4
69	5	・	12-0-9-5
70	6	ヲ	12-0-9-6
71	7	ァ	12-0-9-7
72	8	ィ	12-0-9-8
73	9	ゥ	12-8-1
81	10	ェ	12-11-9-1
82	11	ォ	12-11-9-2
83	12	ャ	12-11-9-3
84	13	ュ	12-11-9-4
85	14	ョ	12-11-9-5
86	15	ッ	12-11-9-6
129	16	ア	12-0-1
130	17	イ	12-0-2
131	18	ウ	12-0-3
132	19	エ	12-0-4
133	20	オ	12-0-5
134	21	カ	12-0-6
135	22	キ	12-0-7
136	23	ク	12-0-8
137	24	ケ	12-0-9
138	25	コ	12-0-8-2
140	26	サ	12-0-8-4
141	27	シ	12-0-8-5
142	28	ス	12-0-8-6
143	29	セ	12-0-8-7
144	30	ソ	12-11-8-1
145	31	タ	12-11-1
146	32	チ	12-11-2
147	33	ツ	12-11-3
148	34	テ	12-11-4
149	35	ト	12-11-5
150	36	ナ	12-11-6
151	37	ニ	12-11-7
152	38	ヌ	12-11-8
153	39	ネ	12-11-9
154	40	ノ	12-11-8-2
157	41	ハ	12-11-8-5
158	42	ヒ	12-11-8-6
159	43	フ	12-11-8-7
162	44	ヘ	11-0-2
163	45	ホ	11-0-3
164	46	マ	11-0-4
165	47	ミ	11-0-5
166	48	ム	11-0-6
167	49	メ	11-0-7
168	50	モ	11-0-8
169	51	ヤ	11-0-9
170	52	ユ	11-0-8-2
172	53	ヨ	11-0-8-4
173	54	ラ	11-0-8-5
174	55	リ	11-0-8-6
175	56	ル	11-0-8-7
186	57	レ	12-11-0-8-2
187	58	ロ	12-11-0-8-3
188	59	ワ	12-11-0-8-4
189	60	ン	12-11-0-8-5
190	61	゛	12-11-0-8-6
191	62	゜	12-11-0-8-7

EBCDIC Number	Kanji Characters Number	Graphic	Punched Card Code
87	1	道	12-11-9-7
88	2	字	12-11-9-8
89	3	残	11-8-1
98	4	地	11-0-9-2
99	5	式	11-0-9-3
100	6	会	11-0-9-4
101	7	支	11-0-9-5
102	8	大	11-0-9-6
103	9	殿	11-0-9-7
104	10	出	11-0-9-8
105	11	商	0-8-1
106	12	方	12-11
112	13	号	12-11-0
113	14	局	12-11-0-9-1
114	15	中	12-11-0-9-2
115	16	小	12-11-0-9-3
116	17	円	12-11-0-9-4
117	18	入	12-11-0-9-5
118	19	部	12-11-0-9-6
119	20	課	12-11-0-9-7
120	21	丁	12-11-0-9-8
121	22	目	8-1
128	23	年	12-0-8-1
139	24	月	12-0-8-3
155	25	日	12-11-8-3
156	26	計	12-11-8-4
160	27	所	11-0-8-1
161	28	社	11-0-1
171	29	店	11-0-8-3
176	30	都	12-11-0-8-1
177	31	株	12-11-0-1
178	32	府	12-11-0-2
179	33	県	12-11-0-3
180	34	市	12-11-0-4
181	35	郡	12-11-0-5
182	36	区	12-11-0-6
183	37	町	12-11-0-7
184	38	村	12-11-0-8
185	39	番	12-11-0-9
192	40	東	12-0
202	41	京	12-0-9-8-2
203	42	済	12-0-9-8-3
204	43	末	12-0-9-8-4
205	44	錠	12-0-9-8-5
206	45	貸	12-0-9-8-6
207	46	借	12-0-9-8-7
208	47	有	11-0
218	48	協	12-11-9-8-2
219	49	注	12-11-9-8-3
220	50	上	12-11-9-8-4
221	51	売	12-11-9-8-5
222	52	買	12-11-9-8-6
223	53	下	12-11-9-8-7
224	54	住	0-8-2
225	55	生	11-0-9-1
234	56	産	11-0-9-8-2
235	57	昭	11-0-9-8-3
236	58	和	11-0-9-8-4
237	59	名	11-0-9-8-5
238	60	阪	11-0-9-8-6
239	61	行	11-0-9-8-7
250	62	資	12-11-0-9-8-2

EBCDIC Number	Kanji Characters Number	Graphic	Punched Card Code
251	63	財	12-11-0-9-8-3
252	64	品	12-11-0-9-8-4
253	65	分	12-11-0-9-8-5
254	66	型	12-11-0-9-8-6
255	67	用	12-11-0-9-8-7

Figure 14.
IBM System/360 Code Assignments in 8-Bit EBCDIC and Punched Card Codes for Katakana and selected Kanji

Source: Ref. 20

Figure 15 shows three examples of Preferred Character Set configurations as documented in the IBM System/360 Product Specifications of 1965. Information is no longer available as to which of these, or other versions were actually materialized.

A Multi Function Card Machine (MFCM), the IBM 2560 was new to the system. This device came with a 5x7 dot matrix wire printer, and included Katakana capability (this technology became the basis for the IBM 2245 Kanji Printer).

IBM Corporation could now offer a variety of Katakana configurations, but this was not enough. Along with the growing usage of computers in the Japanese society, inconvenient restrictions due to the lack of Kanji handling capabilities became more and more pronounced. IBM Japan began to address this unique problem as one of the most urgent company-wide tasks.

Preferred		Characters			Lines per Minute
Set	Repeat	Graphics	Count	Total	
A	4	42 Katakana, 1 Accent, 10 Numeric, 4 Specials	57	228	500
B	3	3 Katakana, 1 Accent	4	12	392
		Total	61	240	
A	6	10 Numeric, 2 Specials	12	72	692
B	2	45 Katakana, 2 Accents, 4 Specials, 15 Kanji	66	132	273
C	1	26 Alpha, 6 Specials, 4 Kanji	36	36	143
		Total	114	240	
A	2	45 Katakana, 2 Accents, 32 Kanji, 10 Numerics, 10 Specials	99	198	273
B	1	2 Kanji, 26 Alpha, 14 Specials	42	42	143
		Total	141	240	

Figure 15. Preferred Character Set Samples   Source: Ref. 20

IBM System/360 Korea

After the war had ended in Korea, a market was to be expected to emerge in its South. Instead of traveling there, I met several times with a young Korean student who was studying in Japan. We analyzed the Korean written language, as needed to determine the technology requirements for data processing. Limiting it to the status of the IBM System/360 technology of that time, I was able to come up with a concept for implementation on RPQ basis. Encoding under EBCDIC and mechanical type slug printing on the IBM 1403 were cumbersome and

yielded a legible, but a somewhat improvised print output. Unfortunately, documentation of this approach could not be located for this publication.

IBM System/360 Thailand

During the mid-sixties, I made several visits to IBM Thailand to analyze that country's national character set requirements. Together with native employees, we defined EBCDIC code assignments for all components of the Thai characters. Consonants, vowels, and tone marks were all placed on discrete code positions (Figure 16). This first Thai EBCDIC encoding was revised twice during the ensuing twenty years, until a Thai National Standard was published in 1986.

Data entry would have to be done via punched cards, with all characters in sequence – top vowels and tone marks followed by consonants followed by bottom vowels. The card codes for the Thai symbols must have been those that were assigned to the corresponding EBCDIC positions. A chart similar to Figure 14 must have been available for an RPQ design of an IBM 026 Keypunch. Likewise, an IBM Card Reader modification must have been implemented.

Together with alphanumeric and specials, it was possible to accommodate all Thai symbol elements within the 240 type slug positions of the IBM 1403 Chain Printer, to be controlled by means of a Preferred Character Set arrangement. In order to print composed Thai symbols - consonants with vowels and tone marks - in the same position where they belong, three print line cycles of the IBM 1403 were taken. As a result, the print speed for Thai text dropped down to approximately one fourth. Allowing the tone marks to appear on the same line with the upper vowels reduced the actually required four levels to three. This compromise was made to avoid that the print speed would drop even further. Figure 17 shows some sample type slug faces, and how they would compose the Thai characters in three print line increments.

The Figures 16 through 18 are copies from original documentation of the mid sixties, provided by IBM Thailand.

	00				01				10				11			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	16	32	48	64 SP	80 &	96 -	112 ฐ	128 น	144 ภ	160 ษ	176 ะ	192	208	224	240 Ø
0001	1	17	33	49	65 ◌ิ	81 ◌๊	97 /	113 ฑ	129	145	161	177 า	193 A	209 J	225	241 1
0010	2	18	34	50	66 ◌ี	82 ◌๋	98 จ	114 ฒ	130	146	162	178 เ	194 B	210 K	226 S	242 2
0011	3	19	35	51	67 ◌ึ	83 ◌์	99 ฉ	115 ณ	131	147	163	179 แ	195 C	211 L	227 T	243 3
0100	4	20	36	52	68 ◌ื	84 ◌้	100 ช	116 ด	132	148	164	180 โ	196 D	212 M	228 U	244 4
0101	5	21	37	53	69 ◌ั	85 ก	101 ซ	117 ต	133	149	165	181 ใ	197 E	213 N	229 V	245 5
0110	6	22	38	54	70 ◌ุ	86 ข	102 ฌ	118 ถ	134	150	166	182 ไ	198 F	214 O	230 W	246 6
0111	7	23	39	55	71 ◌ู	87 ค	103 ญ	119 ท	135	151	167	183 ำ	199 G	215 P	231 X	247 7
1000	8	24	40	56	72 ◌่	88 ฆ	104 ฎ	120 ธ	136	152	168	184 ๆ	200 H	216 Q	232 Y	248 8
1001	9	25	41	57	73 ◌็	89 ง	105 ฏ	121	137	153	169	185 ฯ	201 I	217 R	233 Z	249 9
1010	10	26	42	58	74 ¢	90 !	106	122 :	138 บ	154 ม	170 ส	186 ฦ	202	218	234	250
1011	11	27	43	59	75 .	91 $	107 ,	123 #	139 ป	155 ย	171 ห	187 ฃ	203	219	235	251
1100	12	28	44	60	76 <	92 *	108 %	124 @	140 ผ	156 ร	172 ฬ	188 ฅ	204	220	236	252
1101	13	29	45	61	77 (	93)	109 _	125 '	141 ฝ	157 ล	173 อ	189 ◌ํ	205	221	237	253
1110	14	30	46	62	78 +	94 ;	110 >	126 =	142 พ	158 ว	174 ฮ	190 .	206	222	238	254
1111	15	31	47	63	79 |	95 ¬	111 ?	127 "	143 ฟ	159 ศ	175 ฤ	191 ๅ	207	223	239	255

Figure 16. Code assignments for Thai in EBCDIC, 1964 Source: Ref. 20

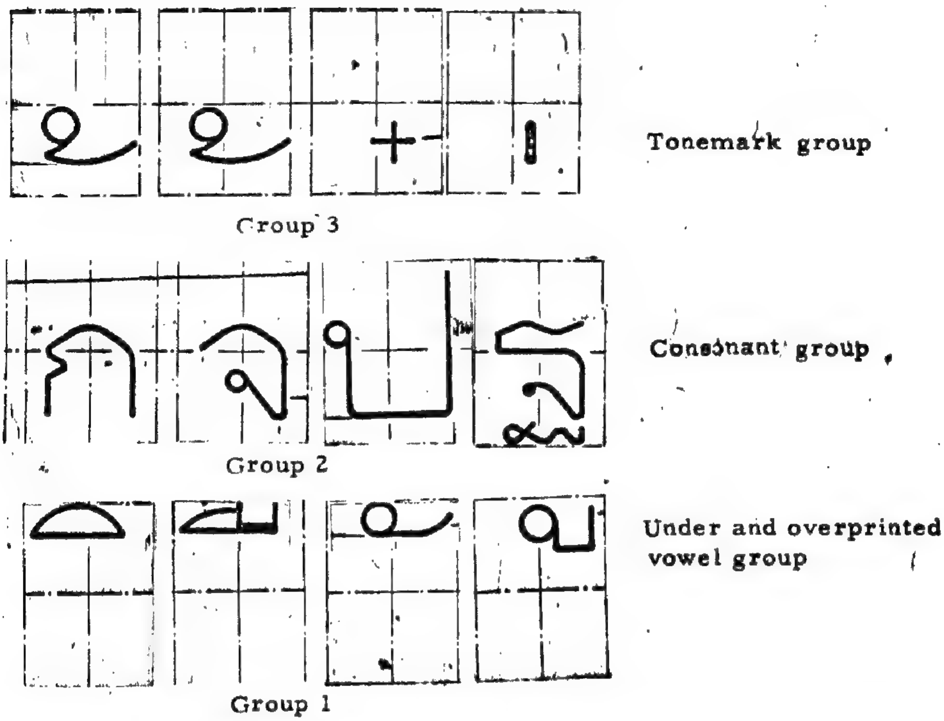


Figure 17. Thai RPQ for IBM System/360, Type Slug Faces Source: Ref. 17

Prior to printing, the serially entered codes had to be rearranged such that the consonants, vowels, and tone marks appear on the same print position. IBM Thailand developed this program. It controlled that a first chain revolution of the printer printed the tone marks and overprinted vowels, followed by a line increment of the paper. The next chain revolution printed the consonants followed by a second line increment. The third line printed the vowels that are to appear underneath the consonants. Thus, one Thai print line required three regular print lines to complete. Figure 18 shows the program instructions, copied from an actual document, in RPG (Report Program Generator) language.

IBM Thailand had no other choice but to live with the resulting lower print speed and wider line spacing, but at least they had a solution to accommodate their national language. The IBM System/360 with Thai capabilities was marketed to an estimated ten sites. Amongst those were the Metropolitan Electricity Authority of Thailand, the Customs Department, the Siam Cement Group, the National Statistics Office, and the Telephone Organization of Thailand.

July 24, 1969.

RPG THAI PRINT SUBROUTINE

RPG SPECIFICATIONS

1. Thai character field in RPG must be called TFLD.

2. User must specify the length of TFLD and output fields by moving the desired length into a numeric field called LNTH with the length of two bytes.

3. The EXIT routine is called THAI and must be specified after the MOVE instruction.

4. Output fields must be named LNE1, LNE2, LNE3 and LNE4 respectively. These fields may have the same length or smaller than the length of TFLD.

5. For printing, there should be no spacing between LNE2 and LNE3, and one spacing between the other lines.

Example of using RPG THAI PRINT SUBROUTINE:

RPG CALCULATION SPECIFICATIONS

Definition	Operation	Factor 2	ResultField	Field Length	Decimal Postn.
Card Column	28 32	33 42	43 48	49 51	52
	MOVE	20	LNTH	2	0
	EXIT	THAI			
	RLABL		LNTH		
	RLABL		TFLD		
	RLABL		LNE1	20	
	RLABL		LNE2	20	
	RLABL		LNE3	20	
	RLABL		LNE4	20	

Figure 18. Thai Print Program

Source: Ref. 17

Another special program was necessary to accommodate the rules for sorting the Thai language. But, despite the described drawbacks, IBM Thailand implemented its national language feature on RPQ basis.

The picture shown in Figure 19 was taken during one of my visits in 1964, with the IBM Thailand General Manager, Chuck Reinbrecht investigating the first prototype of a 1403 Thai Print Train, listening to the comments from Kurt Hensch on a Thai IBM System/360 test printout. Looking on at the center rear is Xua Surathep, IBM Thailand Customer Engineering Manager. He contributed to the hardware development. On the left is IBM Thailand Systems Engineer Pravit Chattalada, who wrote the required Thai programming support.

Figure 19. Investigating the Thai Print Train and Test Print, Bangkok 1964
Source: Ref. 20

Other Far Eastern Countries

Except for Japan, the political situation in East Asia did not present any sizable market potentials in the information technology area during the mid 1960s. Only the Japanese, Chinese, Korean, and Thai languages deemed to be worth consideration.

A limited market for Chinese characters already existed with Taiwan and Hong Kong, and I visited the IBM Branch Office in Taipei to study

their requirements. The branch manager, a veteran IBM employee, literally cried on my shoulder saying that he can no longer read the Chinese newspapers from the communist mainland, because Mao Ze Dong had introduced many so-called simplifications.

I planned, but could not realize a follow-up visit, as it turned out for political reasons. I was denied a visa for a second entry, because I had a stamp in my passport from my visit to an old auntie in communist East Germany, thus ending my effort on Chinese.

I made one visit to the small IBM Branch Office in Burma in 1964 to analyze their national character set requirements. The Branch manager received me with enthusiasm, but there was no follow-up due to the isolated status of that country, and its very limited market potential.

It was impossible to establish the requirements for Viet Nam in the mid sixties, because the war was going on. More than 30 years later, Ken Lunde (see Note below) defined the unique requirements for Vietnamese in his book "CJKV". Viet Nam uses the Latin alphabet, many with accents, in combination with Chinese Hanzi.

Note: Ken Lunde authored "CJKV", a summary of language structures, character sets, and coded character sets standards information on Chinese (traditional and simplified), Japanese, Korean, and Vietnamese (Ref. 2), provides over 1200 pages of complete and excellent guidance for developers of future products. If this had been available in 1961, my work would have been much easier then.

Lunde's book reflects, amongst numerous other topics, the result of many years of studies on coded character sets standards on national as well as international levels. The key standard on Far Eastern Languages, ISO 10646 "Information Technology – Universal Multiple-Octet Coded Character Set" was not published until 1993 by a designated joint committee ISO/IEC JTC1/SC2, between the International Organization for Standardization (ISO) and the International Electro-technical Committee (IEC).

Initial Efforts, Full Kanji in Japan, Early 1970s

Start of Activities

While the available Katakana features were marketed, I was appointed to the position of Development Engineering Advisor to IBM Asia Pacific Headquarters in 1963. My task was to find solutions for implementing the Japanese language in future products, including full Kanji capability. The task encompassed study and analysis of other East Asian languages to the extent necessary to come up with product proposals for the markets in the affected countries.

Realizing that the Katakana features previously made available in Japan were only primitive "band-aid" solutions, I placed first priority on seeking approaches that would facilitate data processing with unrestricted availability of Kanji. Frequent visits to many IBM Development Laboratories in the USA and Europe allowed me to obtain and maintain the understanding and knowledge of new technologies that could possibly be applied in hardware and software.

Obviously, the implementation of any Kanji program required more manpower than just one Engineering Advisor. On my request, IBM Japan provided three qualified individuals to work with me on detailed plans to proceed. Masumi Iwao, hardware engineer from Product Engineering, Hal Nose, a programmer from Sales, and Shun Onuki from Field Engineering. Masumi Iwao was to lead the group called "IBM Japan Engineering Liaison".

Gradually, we were able to narrow down the essential points towards defining answers to the technology requirements for the critical areas. All three men advanced within IBM Japan over the following years, and each one of them had a successful career, by stepping through leadership positions in the Product Development Laboratory of IBM Japan at Fujisawa.

Double-Byte Encoding of Kanji

It had become clear by then that an EBCDIC assignment to 67 Kanji characters in the IBM System/360 as shown in Figure 14 was not sufficient to process the Japanese language. A new binary coding scheme had to be devised in order to provide a sufficient number of discrete code positions for more than 10,000 characters in Japanese, or for Chinese perhaps 30,000.

By looking at the progression in the binary scale, 15 bits could do the trick, because 2 to the power of 15 are 32,768. However, the number 15 did not seem to fit well into the existing computer architecture. One more step in the scale by going to 16 appeared logical.

After all, once a new coding scheme had to be introduced, it should not only provide ample vacant positions for more control functions in future computing systems, but also facilitate the assignment of unique codes to all graphics of all languages in the world. Partially by intuition, the conclusion emerged to go to 16 bits with a double-byte 8-bit EBCDIC concept, yielding 65,536 code points at our disposal.

Superficially, the double-byte concept appears to short-change the Japanese language. This is not so, because each Kanji ideograph has a meaning, which requires several alphabetical characters to represent. Thus, at least two, but in many cases more than two 8-bit EBCDIC would be required to write one Kanji, if using alphabetical characters.

An example is the name of the country itself. The two Kanji 日本 pronounced "Nippon" or "Nihon" are standing for "Japan". The first Kanji means sun, the second means root, both together then mean "rising sun". Thus, six 8-bit bytes would be required to encode "Nippon", while 日本 needs only four. The encoding of Kanji will be explained the section "Kanji Coded Character Set Standardization".

Matrix Dot Printing Techniques Explored

Unless one wanted to provide a mechanical printer with 10,000 type slugs at astronomical cost, rendering microsecond processor speed useless, the only solution was digitization. Each character should be represented in a dot matrix pattern.

In order to ensure professional appearance and quality, I searched for a high-resolution printer within IBM. I found out about a Photo Composer being developed at the IBM System Development Division (SDD) Laboratory in Boulder, Colorado. It was code-named "Pikes Peak", later announced as IBM 2686. It could produce extremely high, superb quality printing. They had a fairly large font library on their host computer, but only Latin characters.

The experts developing the 2686 software, Dieter Paris & Derald Nye, proudly showed me their test prints. It included the large 8-stripe IBM logo, and also two sample Kanji characters. I was impressed. However, I had to give up proposing this machine as a Kanji printer. Its cost, I

believe, exceeded half a million dollars. In addition, there was a strong market requirement for printed copies, as many as six parts, that called for impact printing. Multiple printing by non-impact printers were not thought to be legally acceptable to substitute for transaction copies.

REVISED KANJI PROGRAM PROPOSAL September 16, 1966

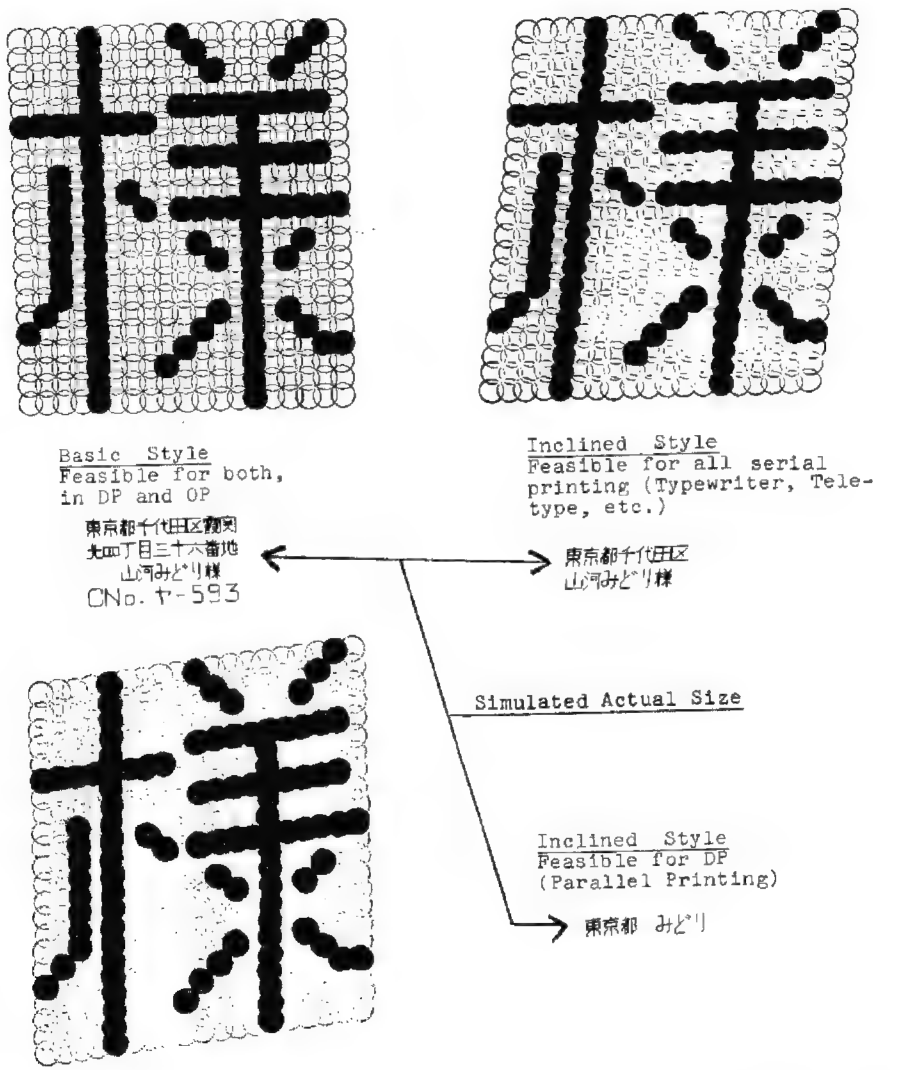


Chart 4. Kanji in Dot Pattern Matrix
(Principle in 20:1 scale, and simulated examples of about actual size)

Figure 20. Simulated Kanji Dot Patterns Source: Ref. 20

Thus, simple, less expensive wire dot printing technology had to substitute, although the quality would suffer. A dot matrix, at a minimum, must have enough vertical and horizontal dots to achieve a resolution that can legibly represent the most complicated ideograph. All other, simpler letters would then be legible automatically. Analyzing the most complicated Kanji from the largest Japanese dictionary revealed a need for a matrix of 19x24 dots. See Figure 20, which is taken from my original IBM document prepared in 1966.

A vertical arrangement of 24 print wires that would sweep over the paper could print the dot patterns of all characters horizontally. By rotating the dot matrix electronically 90 degrees counter clockwise, a print sweep would produce vertical printing that is practiced in Japanese and Chinese, alternating with horizontal. In order to provide this feature, the dot matrix should be made a square shape of 24x24.

Although the 24x24 dot matrix could print all Kanji, the quality would be at a minimum. In order to improve this, two vertical rows of twelve print wires each could be arranged, partially offset against each other. Slanting the print wire arrangement slightly could produce an italic writing style. Figure 21 illustrates this concept. This picture was also part of the original IBM document of 1966 entitled “Proposal for a Kanji Program”.

As a possible wire dot printing candidate, I initially proposed to make use of the magnetostrictive technology, which had been developed in the IBM Germany Development Laboratory by Max Preisinger. His prototype with seven wires was working well. However, the idea was dismissed because of the high energy requirement to achieve wire elongation. A 24-wire print head would be bulky, causing a prohibitive power consumption.

As a result, Kanji printing in a dot matrix using wire print heads had to wait until the wire print technology developed by the IBM Endicott Development Laboratory for the IBM 2560 MFCM could be redesigned and adopted for the IBM 2245 Kanji Printer later. This wire impact printing approach employed individual sets of tiny magnet and armature assemblies, each of which could be driven independently by an electronic impulse. A fine wire was bonded to each armature piece so that the pull and release motion of the armature could directly drive the wire against the ribbon in front of a sheet of paper. This technique allowed flexibility for forming any character in a dot matrix representation.

REVISED KANJI PROGRAM PROPOSAL September 16, 1966

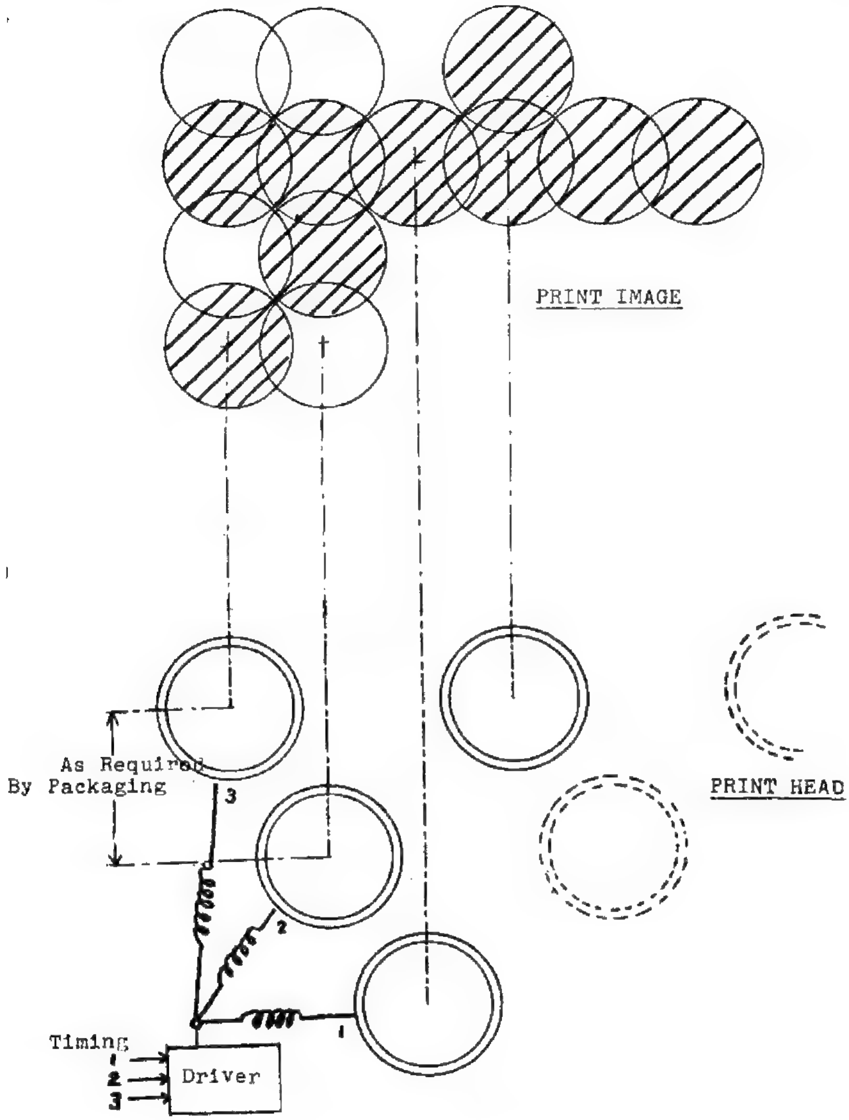


Chart 5. PRINCIPLE OF "OVERLAPPED PRINTING"

Figure 21. Overlapped Dot Pattern Concept Source: Ref. 20

Storage Requirements for Kanji

As a minimum, a computer with full Kanji capability had to have a fast access memory with at least as many 8-bit double-bytes as there are

characters, let's assume 10,000. This meant 20K EBCDIC code positions already. But it is only the lower limit of the requirement, because the dot pattern matrix of each character, associated with its code had to be stored as a font library.

Now it was possible to compute the total storage space requirements. By multiplying the 24x24 dot matrix with the 10,000 characters, a storage requirement of 5.76 Megabits emerged, being approximately 720 KB of 8-bit EBCDIC.

Magnetic core memory was the technology of the mid 1960s, but at a premium cost. Thus, we faced a dead end street with such a vast and fast access memory requirements. Paper tape, magnetic tape or punched cards would slow down the data input/output and processing speeds prohibitively. We did not consider punched cards in order to avoid the need for defining card codes and equipment.

At this time, IBM introduced a new technology magnetic hard disk for the IBM 1620 Scientific System. Massi Iwao, Shun Onuki, and I went to a customer's installation to see it, and we determined it could be a possible candidate for an electronic Kanji type font library.

Kanji Data Entry by Phonetic Grouping

Data entry for thousands of different characters was the biggest nut to crack. The mere thought to have to go via punched card codes was a nightmare. Needed was a method to produce, and enter the double-byte EBCDIC codes directly into the computer. However in 1966, no suitable computer terminals were available, nor were all-points-addressable monitors.

This led me to propose a solution, which the Patent Department of IBM Japan documented as a candidate for protection. The device would have the shape of a conventional typewriter with a wire dot pattern print head and a keyboard, which is also the monitor. All characters would be projected from microfilm frames below to translucent key tops as shown in Figure 22.

In home mode, the key tops would display the basic set of Hiragana characters (Sets 1 and 3 of Figure 10), which would fit onto the conventional 48 keys. The basic idea for data entry was to find the group of Kanji by pronunciation with the first keystroke. This would move the microfilm to the frame from which all Kanji with that pronunciation

could be projected onto the key tops. Figure 23 shows some sample Kanji groups on the microfilm.

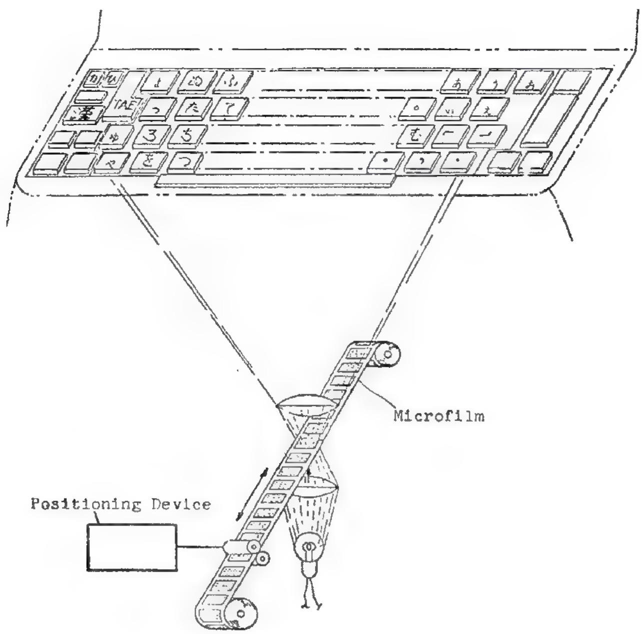


Figure 22. Kanji Data Entry Keyboard Source: Ref. 20

Most Kanji groups of the same pronunciation have less than 48 ideographs. Therefore, the vast majority of Kanji could be identified by two keystrokes, which are needed for producing the two EBCDIC bytes that are allocated to the selected Kanji via an electronic translation table.

The top frame of the shown film segments carries the basic Hiragana set used to select the Kanji by its pronunciation. The four frames below the Hiragana group are Kanji group examples, the pronunciation of which starts with "A", the next "I", then "U", followed by "E".

A cross-section view of the proposed device is shown in Figure 24. It could be developed into a Kanji Typewriter, Teletype, and also function as a computer terminal. The arrangement of the translucent key tops (labeled "2") with contacts and their controls is illustrated in Figure 25.

This device could have become a candidate for the much needed, but at that time non-existent all-points-addressable Cathode Ray Tube type

of workstation, making punched cards superfluous. Figure 26 shows the overall schematic and logic of the proposed device.

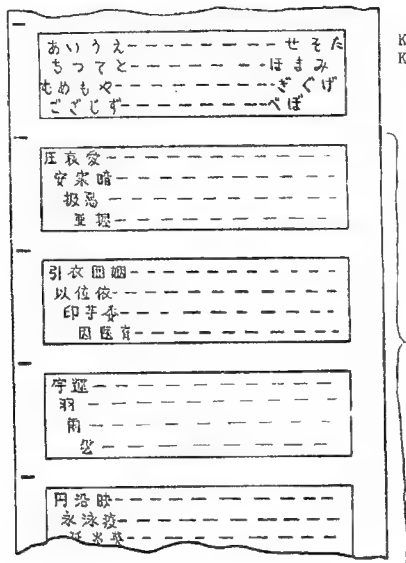


Figure 23. Kanji Groups on Microfilm Source: Ref. 20

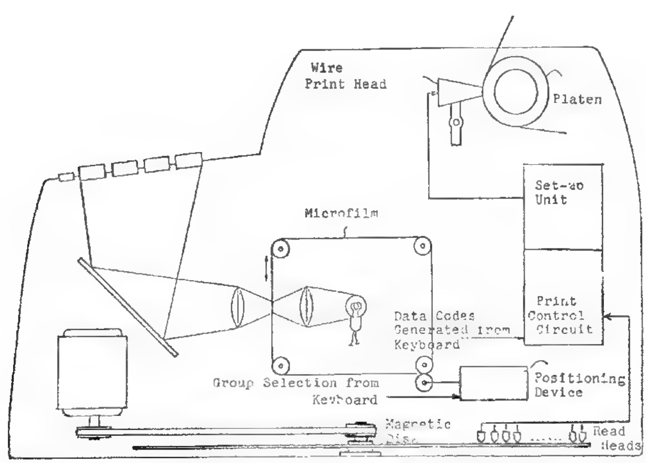


Figure 24. Principle of Kanji Workstation Source: Ref. 20

Note: Figures 20 through 28 are reproductions from illustrations in the original IBM document "Proposal for a Kanji Program" of 1966.

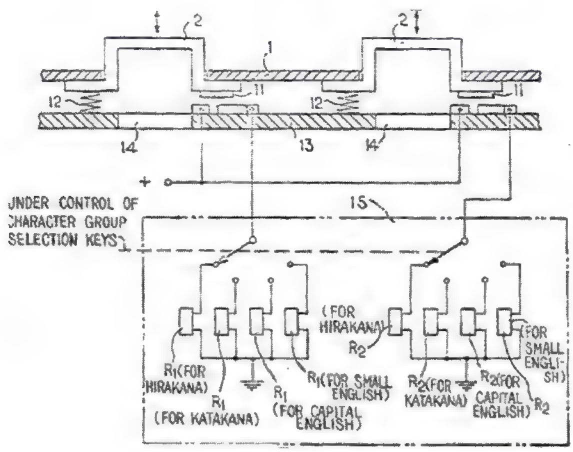


Figure 25. Kanji Terminal Key Top Contacts Source: Ref. 20

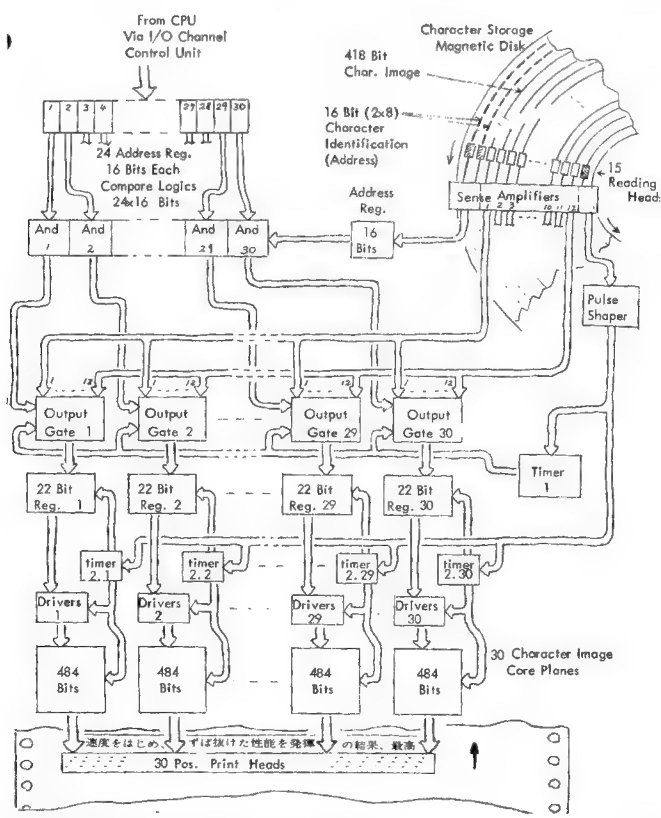
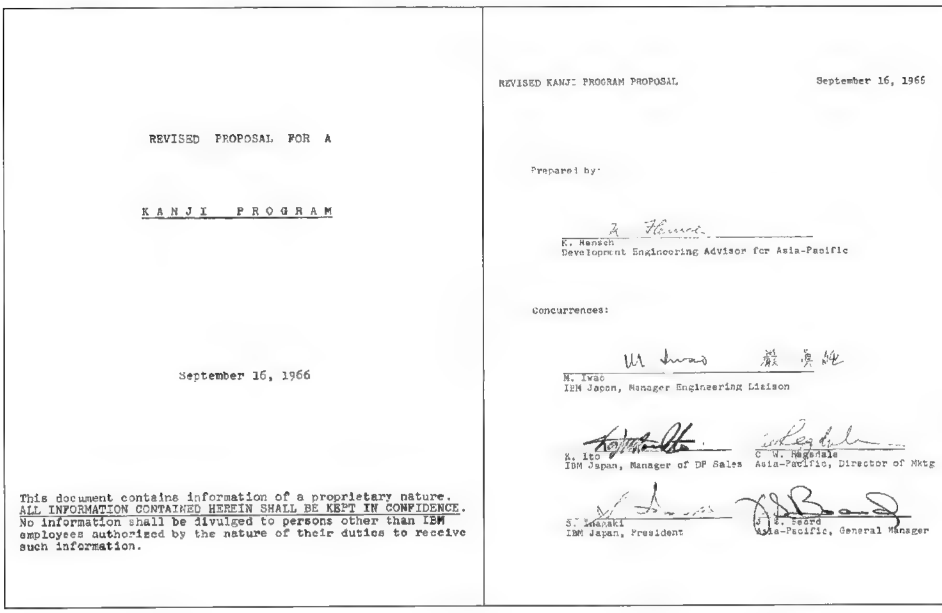


Figure 26. Kanji Terminal Control Circuits Source: Ref. 20

IBM Kanji Program Proposal

I published the result of our work in an IBM internal document dated September 16, 1966 with the title "Proposal for a Kanji Program" (Figures 27, 28). It was the first of its kind, specifying solutions to implement Kanji in future IBM products.



REVISED PROPOSAL FOR A

K A N J I P R O G R A M

September 16, 1966

This document contains information of a proprietary nature.
ALL INFORMATION CONTAINED HEREIN SHALL BE KEPT IN CONFIDENCE.
No information shall be divulged to persons other than IBM employees authorized by the nature of their duties to receive such information.

REVISED KANJI PROGRAM PROPOSAL September 16, 1966

Prepared by:

K. Hensch
Development Engineering Advisor for Asia-Pacific

Concurrences:

M. Iwao
IBM Japan, Manager Engineering Liaison

K. Ito
IBM Japan, Manager of DP Sales

C. W. Ragsdale
Asia-Pacific, Director of Mktg

S. Inagaki
IBM Japan, President

J. [illegible]. Beard
Asia-Pacific, General Manager

Figures 27/28. First IBM Document for Kanji, Title and Sign-Off Pages
Source: Ref. 20

Staff Expansion for Kanji Program

As early as during my work with Katakana in 1963, Lowell Ravesloot, IBM World Trade Corporation Director of Standards asked me to assume the responsibility as consultant for "Corporate Coordinator for Non-Latin Character Sets". It was a side assignment, but turned out to be the beginning of my career in the IBM Standards organization later.

At the end of 1966, I was assigned to Program Director of Standards IBM World Trade Asia. Massi Iwao with Toshiaki Igi and their team took on the lead for the Kanji program. The group quickly expanded and worked diligently towards developing the first IBM system with Kanji capability. They implemented the concept that I had documented in my 1966 "Proposal for a Kanji Program". In less than three years, these young men achieved the realization of the first IBM Kanji Data Processing System.

First Japan Kanji Capability Realization Early 1970s

First IBM Kanji System Appearance in Public

As an example of innovative IBM technology, a Kanji Data Processing System was demonstrated at the EXPO 1970 World Fair in Osaka. It was the first of its kind on the market. The main feature of the system was a Kanji Printer (to be announced a year later as the IBM 2245), which created considerable surprise and attention amongst customers and competitive manufacturers.

Character Encoding in Double Byte EBCDIC

The Kanji Printer operated with a unique internal System/360 Kanji code set. IBM Japan assigned 16-bit codes (double byte EBCDIC) to 3600 characters. All Kanji code assignments at this point of time were arbitrary, because neither usage statistics, nor a national standard code structure for Kanji existed.

The set included 2950 commonly used Kanji, but also Katakana, Hiragana, and alphanumeric and necessary special characters. Katakana came in two versions, half size (single EBCDIC) and full size (double byte).

Wire Dot Printing

The development of the Kanji printer had been assigned to the Product Development Laboratory of IBM Endicott, NY. The chosen technology was wire dot printing, based on the IBM Kanji Program Proposal of 1966, in a matrix of 18x22 partially overlapped dots. Figure 29 shows samples of Kanji and other letters, which this printer was able to produce, and as they appeared in the EXPO 1970 exhibit pamphlet.

The two top illustrations on the right in Figure 29 are of an 18x22 dot matrix, used for horizontal printing. The two upper illustrations in the left column are of an 18x18 matrix, used for vertical printing which are sufficient for the Kanji shown. However, for some very complex Kanji, the 18x18 dot matrix resolution would not be fine enough, including some of the selected 3600 characters of this printer.

Figure 29 also shows that the EXPO 70 Kanji printer could print the existing one byte EBCDIC character set (alphanumeric and Katakana) in a 7x9 dot matrix as well as Kanji on the same print page. This was to demonstrate the Kanji printer's capability of printing characters of the existing alphanumeric system.

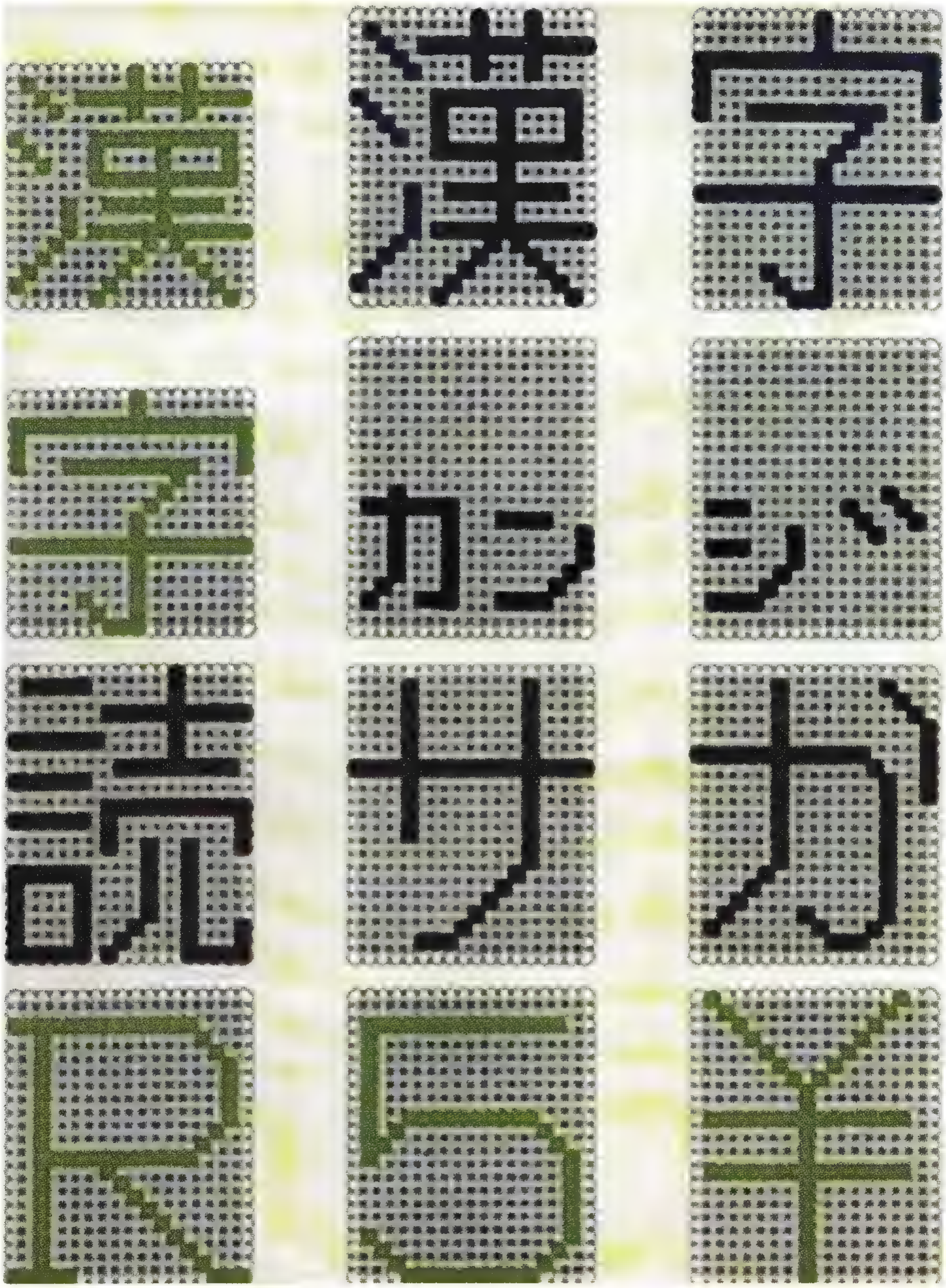
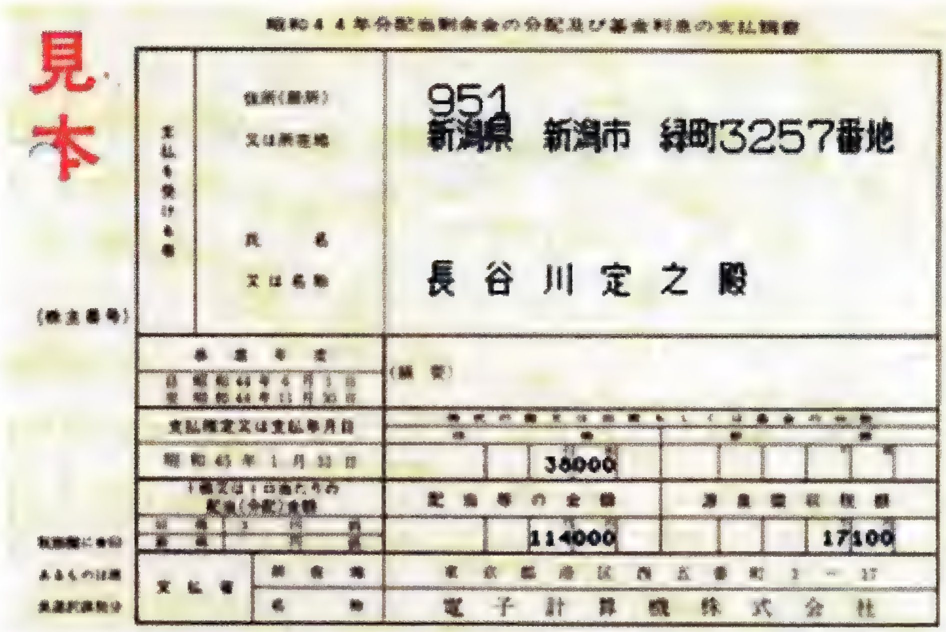


Figure 29. Kanji Printer Dot Pattern Samples

Source: Ref. 14



↑ Paper Movement Paper Movement ↑

Figure 30. EXPO 70 Kanji Print Sample Source: Ref. 14

Figure 30 is the copy of a print sample from the printer exhibited in Osaka in 1970. The printer featured 16 print positions with large characters of 4.6 x 5.8 mm, about double the size as those of the IBM 1403 Chain Printer. Print speed was 300 lines per minute (LPM) at 3 lines per inch, or 333 LPM at 4 lines per inch. In addition to horizontal printing, vertical was also possible (Figure 31).

The wire printing method of the IBM 2560 MFCM was used. The wires were stationary, arranged horizontally over the 16 print positions, which required 18x16=288 wire driver circuits and mechanisms. Vertical wire dot positions were generated by a stepping motor controlled paper feed motion. The 18x22 dot matrix for horizontal printing (Figure 30) was formed by 22 paper feed steps. The 18x18 dot matrix for vertical printing (Figure 31) was done in 18 paper feed steps, by a 90 degrees counter clockwise rotated print image utility program

Ideally, it would have been more economical to design a horizontally moving print head. Such would have consisted only of two vertically arranged columns of wires (eleven each) that swept over the paper. However, the IBM 2560 MFCM print head wire driver mechanism was

not flexible enough to be incorporated in a design that could move sideways over the 16 or more print positions.

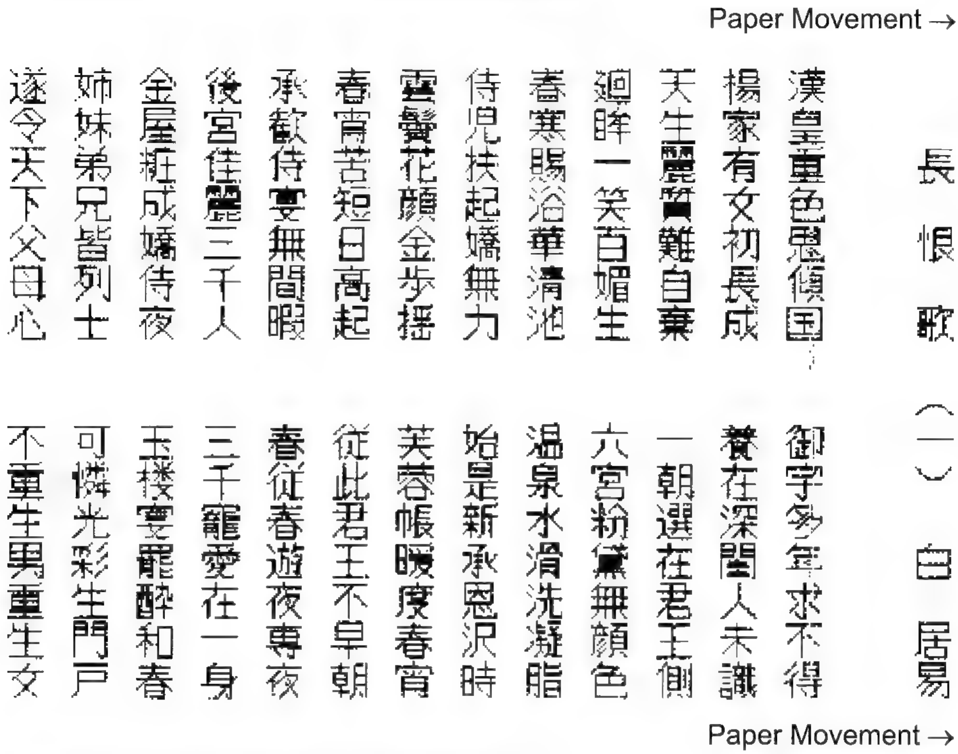


Figure 31. Vertical Kanji Printing Samples Source: Ref. 14

A Mechanical Kanji Typewriter on the Market

We looked at a non-IBM traditional typewriter to study existing Kanji data entry methods (Figure 32). The bright-color lower part is a template with 3600 characters printed on it. The dark, rectangular part above it is a tray filled with mechanical type slugs. Both template and tray are stationary. A platen assembly is mounted on the same base but is movable from right to left in increments of characters.

The small assembly above the slug tray is suspended in a mechanism, which can be moved left, right, back and forward, in order to permit pointing a stylus to any character on the template below to select a character. Once positioned, the U-shaped brace must be pressed down forcefully, causing a grabber to lift the corresponding type slug and to fire it against the platen. The typing speed is very slow, for a trained typist ranging in the neighborhood of 5-10 characters per minute.

Figure 33 shows such a template, reduced and rotated 90 degrees counterclockwise. The character arrangement on the template has a certain order. Despite its poor quality, one can see that the Kanji characters are arranged in pronunciation groups. On-yomi is basically used to determine the Kanji group, but not always. Some usage-driven Kun-yomi Kanji are included. The template also includes Hiragana, Katakana, Roman alphabet, numerals and special characters, plus empty side rows for type slugs of unique Kanji that are not on the template.

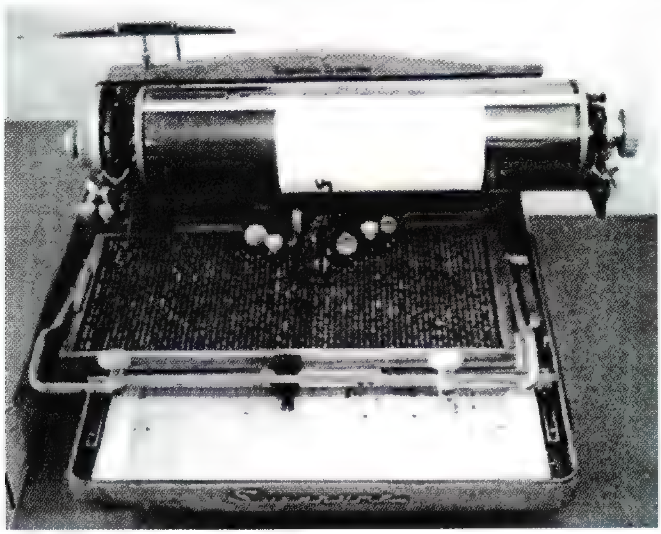


Figure 32. Mechanical Kanji Typewriter Source: Ref. 20

Data Entry Using Complex Keyboards

For Kanji data entry, the IBM Advanced Systems Development Division (ASDD) at Mohansic, NY developed a prototype keyboard, modeled after the template of an old traditional mechanical Japanese Kanji typewriter, shown in Figure 33. IBM did not show this prototype publicly, but used it internally for human factor studies of Kanji data entry.

ASDD engineered the template on the layers of a diaphragm switch arrangement. Pressing a stylus on the image of the desired character closed a corresponding circuit, which in turn generated two EBCDIC punched card codes on a modified IBM 029 Card Punch.

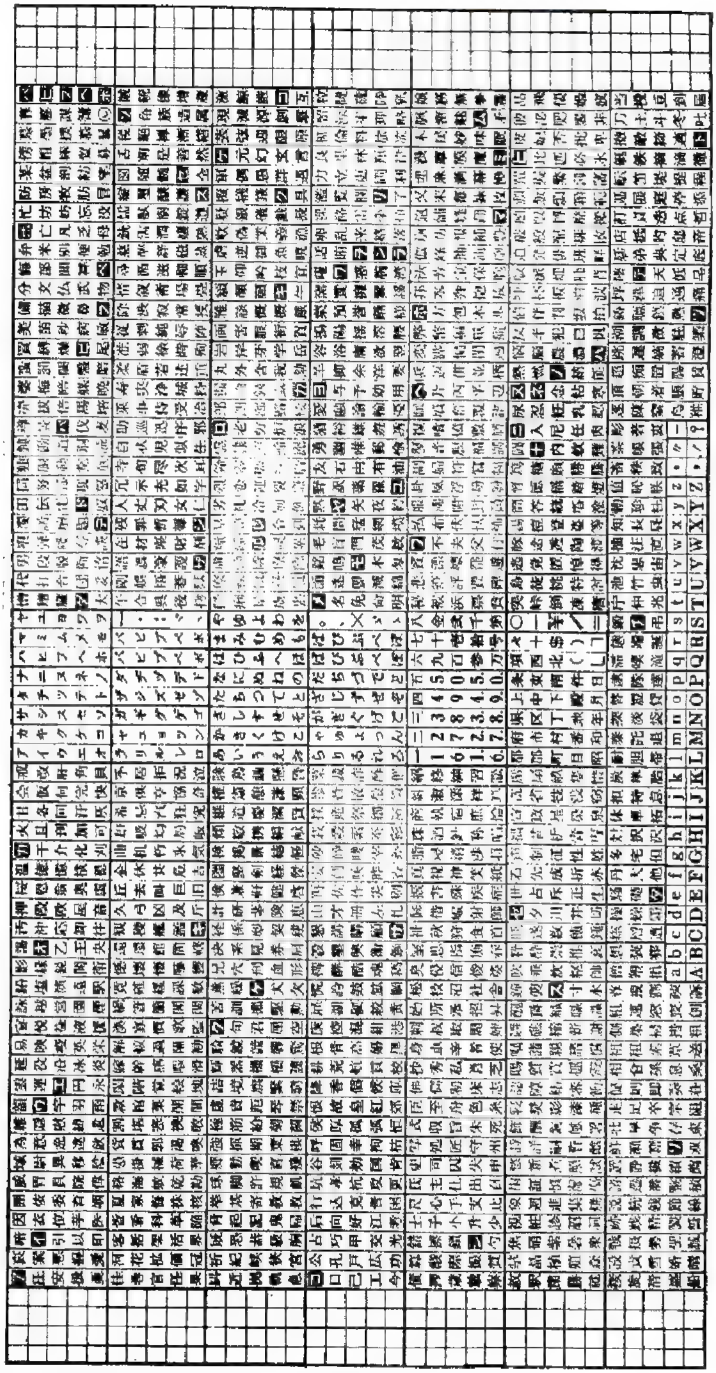


Figure 33. Kanji Typewriter Template Source: Ref. 20

Ideally, though, two EBCDIC codes could have been generated directly, without requiring the interim punched card media. A direct data entry method of key touch to memory had to wait many more years, however, until memory in large enough capacities and low cost became available. Also, keyboard devices directly connected to a CPU were not available at this time. Data had to be read into a CPU via a conventional card reader.

Official IBM Kanji System Announcement

IBM Japan officially announced its first Kanji System in 1971 as a general product. The system consisted of

- An IBM 2245 Kanji Printer directly attached to a System 360/370 Multiplexor channel. This printer (Figure 34) was the centerpiece of the system, and it was the very machine that had been exhibited at the EXPO 1970 World Fair in Osaka.

- An IBM 029 Keypunch with Kanji Keyboard for Kanji data entry to IBM cards (Figure 35).

- A unique package of IBM System 360/370 Operating System (OS) programming support.

The system was capable of handling a set of up to 10,000 Kanji characters. Among these, 7525 were defined as the IBM 2245 Kanji printer character set and a character set manual was provided, presented in a dot matrix pattern. The remaining positions were available for users to design and register additional Kanji, which were not included in the IBM 2245 Kanji printer character set.

Since Kanji are ideographs of an unlimited number, the flexibility of defining user optional characters is mandatory in order to make a Kanji system practical for use. This concept of character set flexibility, established by the IBM 2245 Kanji System, was implemented by all succeeding IBM Kanji system offerings.

IBM 2245 Kanji Printer

See the previous section “Initial Kanji Capability Realization in Japan“. On another US assignment, Masumi Iwao coordinated development and announcement of the IBM 2245 Kanji Printer (Figure 34) with Product Line Management (PLM) of IBM Japan.

Figure 34. IBM 2245 Kanji Printer Source: Ref. 14

IBM 029 Kanji Keypunch

The 15-shift keyboard of the IBM 029 Keypunch (Figure 35) featured 240 keys marked with 15 Kanji each. The left part of Figure 36 illustrates the cluster of 15 shift keys that are located on the left side of the keyboard. The right part of Figure 36 is the enlarged image of one of the 240 keys.

Using the right hand to hold down the key that includes the desired Kanji, the operator used the left hand to press the corresponding number key (6) of the shift key cluster. This dual-action resulted in punching the two columns of EBCDIC card codes identifying the character 都.

The IBM 029 Kanji Keypunch came with 3600 defined keyboard positions, which had the following breakdown:

- 2950 Kanji characters,
- 405 non-Kanji characters (alphanumeric, Hiragana, Katakana, and special characters),
- 145 open for user option additions,
- 100 with double digit numbers from 00 to 99.

Consecutively pressing two of those 100 double digit number keys could generate any character reference number between 0000 and 9999. Since the IBM 2245 Kanji printer system character set support could handle up to 10,000 characters, all 3600 characters could theoretically be entered using this method. However, the primary reason for providing this feature was to facilitate the data entry of user option defined characters, which were not included in the basic keyboard layout.

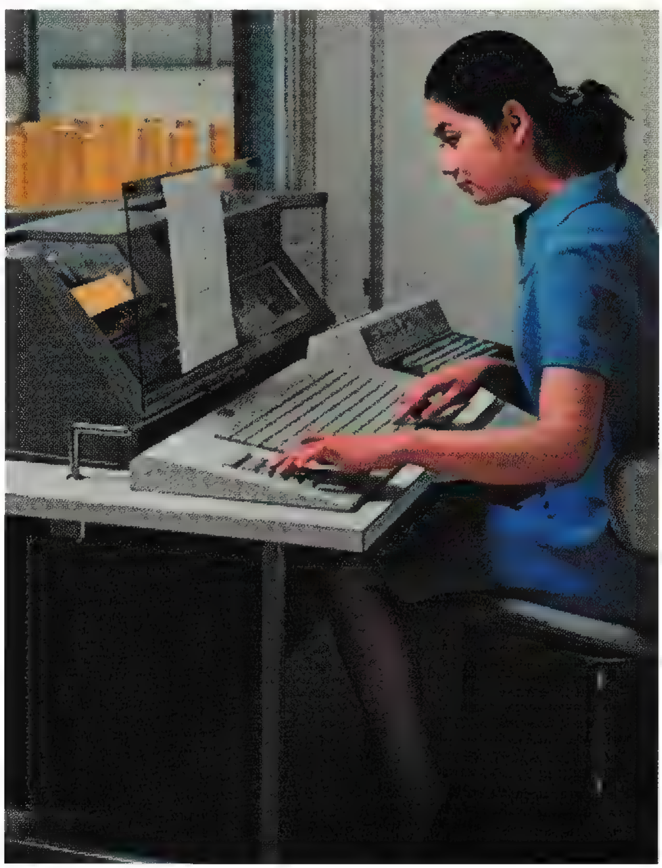


Figure 35. IBM 029 Kanji Keypunch Source: Ref. 14

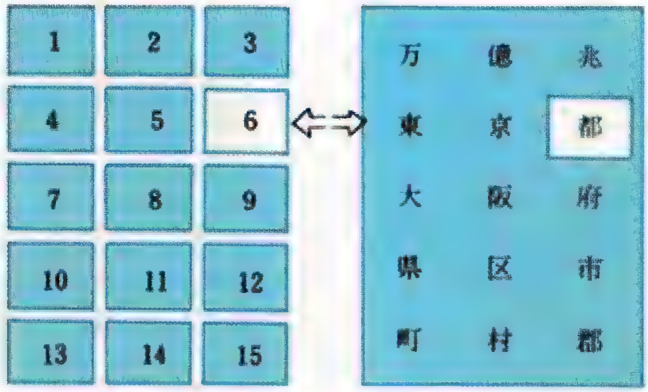


Figure 36. 15-Shift Key Selection Concept Source: Ref. 14

Kanji Programming Support

A "Matrix Character Generation Program" (MCGP) was provided for the System 360/370 Operating System and Disk Operation System (DOS). It had a character set processing capability of up to 10,000 Kanji characters. Two bytes of EBCDIC code (16 bits) were used to encode one Kanji character. Each Kanji character was assigned a unique character reference number ranging from 0000 to 9999. This character reference number was mapped to a two byte EBCDIC binary expression as internal system character code. The programming support provided the following functions:

- Conversion of IBM 029 Kanji card punch code to internal system Kanji character code;
- Generation of an IBM 2245 Kanji Printer font library on the hard disk drive (an 18x22 dot matrix font for horizontal writing Kanji, an 18x18 dot matrix font for vertical writing Kanji, a 7x9 dot matrix font for a one-byte EBCDIC character);
- Design and registration of user defined Kanji character font into the Kanji printer font library;
- Retrieval of character dot matrix pattern from the Kanji printer font library using the two byte EBCDIC code as index key and store in the print buffer;
- Scanning the character dot matrix pattern in the print buffer in synchronization with the printer paper movement and sending these data to the printer over the System 360/370 channel.

Changes of Personnel Assignments

By 1969, I had been relocated to the USA to assume an assignment with the Standards organization of IBM World Trade Headquarters in New York. I was no longer directly involved in subsequent developments to support East Asian languages in IBM products. Several years later, however, I actively participated again in the publication of IBM Internal Standards for Japanese, Chinese, and Korean Coded Character Sets (see the Code Standardization section).

Soon thereafter, Massi Iwao assumed different responsibilities, and Toshiaki Igi was promoted to manager of the group for linguistic support. His team was soon incorporated into the Product Development Laboratory of IBM Japan, which had just been established at Fujisawa, West of Tokyo. For many years, Toshiaki Igi spearheaded the developments in support of national languages for IBM products.

IBM Kanji System Reference Manuals

National standardization still years away, IBM Japan published in April 1972 a comprehensive Kanji Character Set and Code Reference Manual (N:GA18-1018) for users of the IBM 2245 Kanji Printer System (Ref. 22). It was a first ever of its kind. This manual defined 7525 double byte characters (3355 characters for the Kanji keyboard and 4170 characters beyond the Kanji keyboard set), plus 127 one-byte characters with individual character reference number assigned. The manual was later upgraded for the IBM 5550 as N:GC18-2040 (Ref. 23), and for following IBM Kanji systems as N:GC18-0611 in August 1979 (Ref. 24), and N:GC18-0785 in 1986 (Ref. 25).

The majority of Kanji system users kept referring to these IBM Kanji Character Set and Code Reference manuals. These manuals have played an important role in the later development of Kanji Coded Character Set Standards.

An incredible effort was required to decide which Kanji should belong, and then publish them as a character set manual. The first step was to gather as many Kanji characters as possible to begin with, amounting to over 60,000. Nobody knows the exact number. Proper identification of every character was necessary, looking up “one by one”, using several reputed Kanji dictionaries.

The next job was the difficult and very time-consuming task of grouping the characters into seven known Kanji categories:

1. Shinji (newly defined everyday use character);

2. Kyu

3. Kyuji (old character);

4. Kyu Honji (authentic character);

5. Kyu Koji (ancient character);

6. Kyu Zokuji (character simplified by daily use);

7. Kyu Goji (undesirable, mistaken character, yet still in use);

8. Kyu Bettaiji (character of same pronunciation and meaning, but from another origin).

This one-by-one examination of all these Kanji characters was required in order to decide which characters should be included in the IBM 2245 Kanji character set. In most cases, consultation and review with the National Institute for Japanese Language was necessary.

This being the first publication of Kanji in a dot matrix pattern presented another challenge. Some very complex Kanji characters were simplified in order to fit into 18x22 or 18x18 dot matrices. Heated discussions took place whether IBM could introduce incorrect Kanji in the manual or not. In conclusion, Kanji characters in dot matrix presentation were printed along with the same character printed by type slug, as reference to the correct Kanji character. All simplified Kanji in dot matrix pattern were noted as simplified character in the manual.

Newspaper Publishing Systems in Japan

Revolutionizing Newspaper Composing

Simultaneous with IBM Japan's effort that culminated in the announcement of a commercial Kanji system with the IBM 2245 Printer, the development of the world's first computer system for newspaper industries was launched in 1971. The objective was to computerize the manual-mechanical systems of Asahi Shimbun and Nihon Keizai Shimbun, two major Japanese newspapers.

Asahi's system was called "NELSON" (New Editing and Layout System Of Newspapers).

"ANNECS" (Automated Nikkei Newspaper Editing and Composing System) was the acronym for the Nihon Keizai Shimbun system.

These systems ran on application programs developed by IBM Japan, called JPS/OLS (the Japanese Publishing System / On-line System), supplemented by peripheral programs developed by the two newspaper companies. The JPS/OLS programs alone consisted of approximately 700 program sections totaling about 93,000 lines of code. The whole program size of the NELSON and the ANNECS each reached more than 140,000 lines of code.

These systems revolutionized the newspaper composing work, which previously involved a large number of lead type slugs that were very noisy. Consequently, newspaper editing and printing was drastically converted from a "hot method" to a "cold method".

Conventional Kanji Entry Methods

Initially, whole newspaper articles had to be entered by punching hole patterns into paper tape using a "Kanji Teletype", devices that were used in the newspaper industry. No word processing was involved at the data input stage. The keyboard of the Kanji Teletype was a multi-shift keyboard similar to the IBM 029 Kanji keyboard of the IBM 2245 Kanji System. Both, the Asahi Newspaper Company and Nikkei Newspaper Company implemented unique Kanji Teletype equipment for their own use. The character set, keyboard layout, and number of keyboard shifts were different between the two.

Conventional paper tape readers read those punched paper tapes, and the signals were converted to internal EBCDIC code. The Kanji code

assignments for their systems were Asahi's and Nikkei's own, because a Japanese Industrial Standard (JIS) did not exist until much later.

Thereafter, an editing task was performed interactively on "Image Display" devices, driven by an IBM 3033 Processing Unit. Tailor-made newspaper editing programs, developed in close collaboration with the participating teams, became the style of software support.

Subsequently, the full-page information was sent to a photo-printing system for creation of a photo-image film. The film was mounted on an automatic print master-plate producer for printing the newspapers. Also, it was sent to remote sites via facsimile for printing of identical newspaper copies elsewhere.

Collaboration in Development

Beginning in 1971, development of the systems was achieved over six years of close collaboration between IBM Japan with the two newspaper companies and several divisions of the IBM Corporation. This teamwork between IBM and the customers was the key factor for successful introduction of the newspaper systems.

The Systems Development Department of IBM Japan, together with the Federal Systems Division (FSD) of IBM Corporation undertook the programming of the JPS/OLS.

FSD also assisted in the development of the IBM 4510 Digital TV (DTV), operated by a regular alphanumeric keyboard and joystick. Instead of seeing the actual Kanji, square symbols presented headlines and mere lines simulated the text on the monitor in the first stage systems. The page layout operator would compose the full page by arranging the square symbols and text lines in accordance with the layout chart made by the layout editor.

Only one font of 16x16 dots was available for the IBM 4510 DTV at the initial stage. Another model of the IBM 4510 digital TV was developed which attached a Kanji Teletype keyboard for Kanji Text editing. An engineer from the Nikkei newspaper company visited FSD and provided technical help for the development of this keyboard attachment to the IBM 4510 DTV. However, the transition to displaying actual characters did not occur in layout editor mode until the year 2000. From that time, the layout monitor was able to present the whole page in characters instead of square symbols and simulation lines.

The Systems Development Division (SDD) of IBM Corporation contributed to improvements to the IBM 2680, a high-speed photo composer that was used in the early stage of the Japanese Publishing System. The Asahi Newspaper Company technically assisted SDD in the development of the IBM 5981, a scanner/plotter called FPPS (Full Page Photo Printing System), used in the later stage of JPS.

The challenge towards the development of the newspaper systems in this early, rather premature stage for both newspaper companies and the IBM project teams turned out to be a showcase for Kanji as well as photographic image processing. The effort was rewarded with success thanks to the high technical capabilities and determination of the participating international members.

Japanese language text processing capabilities were practically unavailable on the market in that time period. It took at least another decade to see Personal Computers appear everywhere that were powerful enough to achieve what the large Asahi and Nikkei systems had been set up to do.

Debut of Computer Generated Newspapers

The two newspaper companies actually employed prototype versions of the systems for printing some editions at the end of December 1971 through January 1972, and made a strong appeal to the public for the use of "computer generated newspapers". Figure 37 is an illustration of the NELSON system network of over three decades ago。

In 1975, the Full-page Photo Printing System (FPPS) was introduced, capable of producing one complete newspaper page via computer editing, and of preparing the master film for printing. Taking advantage of FPPS, the Asahi Newspaper Company made a complete transition to the NELSON system in 1980 for newspaper editing and printing. September 24 saw the first newspaper printed on this publishing system

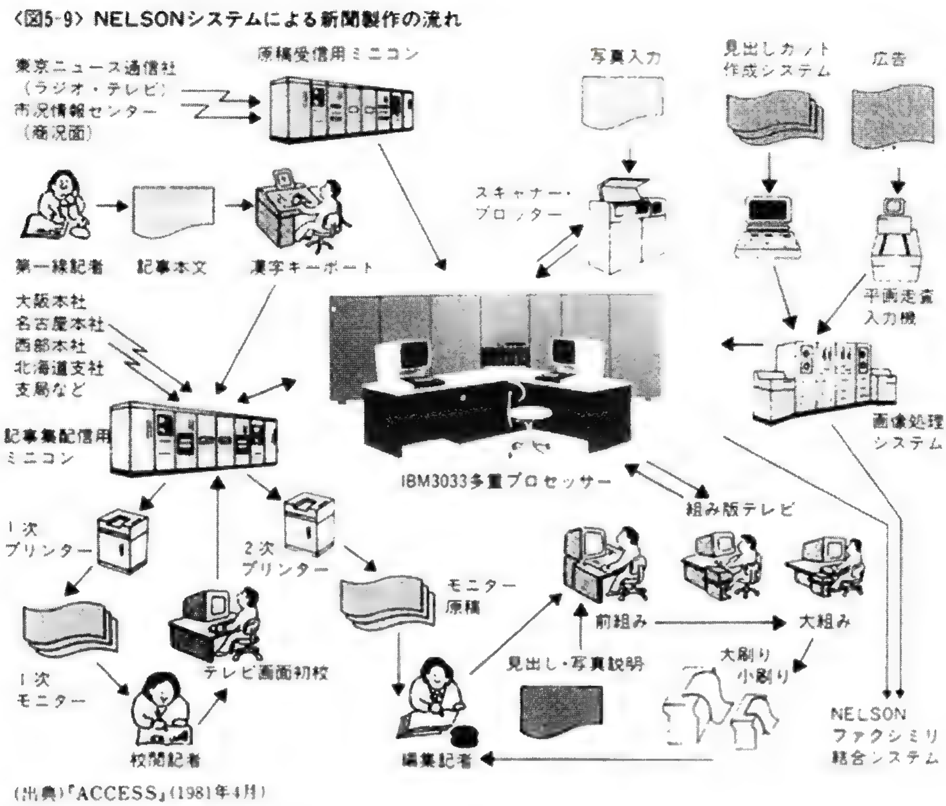


Figure 37. NELSON System Network (Ref. 1)

IBM Japan Taking the Lead, Achievements through 1990s

Limited Capabilities of Initial Products

Although the first emergence of computerized Kanji in the seventies was a breakthrough, it did not satisfy the entire market. At the high end, newspaper-publishing systems were developed to serve their specific purpose. But, they required a large amount of equipment and software at considerable expense affordable only to large companies.

The IBM 2245 Kanji Printer in 1971, a first step for general-purpose products, could not adequately meet all market requirements for Kanji data processing. The small number of single dot-pattern type font print positions, and the cumbersome data entry via punched cards limited the chances for a wider acceptance.

The cost for providing sufficient memory to accommodate more varieties of Kanji characters and sizes would have been prohibitive at that time. Further, insufficient printing technology did not allow providing more characters per line.

IBM Japan through its daily sales activities, however, was continuously under a strong demand pressure from customers for Kanji data processing. It was decided to motivate IBM Corporation towards making Kanji capabilities a standard feature in all its future products, rather than developing them as special features like the IBM 2245 Kanji Printer. Thus, a formal request to the product development divisions of IBM Corporation was issued.

First Step to Meet Market Requirements

Due to the rising market needs and door-opening new technologies in hardware and software, the IBM Corporation decided to establish a product development laboratory in Fujisawa, west of Tokyo. It started with 92 members initially with competence and experience in developing custom products for the Japanese market, which included handling Far Eastern languages in computing.

Kanji Linguistic Study

During 1972/73, Fujisawa Development Laboratory of IBM Japan, which founded in 1971, conducted several linguistic studies on the Japanese language in order to find parameters that could facilitate improvements in system performance and human factor aspects.

- Analyze the character usage frequency to help minimize the access time for identification of a character on the keyboard layout during data entry.
- Define the keyboard layout to reflect the most natural, and most commonly used way of finding a character, whether by stroke pattern or by pronunciation with subsequent selection from groups, based on the Hiragana sequence, alphabetical, or any other methods.
- Determine the optimized data entry method (keyboard with or without shift keys versus stylus versus stroke pattern versus Katakana mnemonic input).
- Compare the efficiency of Kanji selection via pronunciation of individual characters versus lookup from an electronic dictionary. The latter often results in multiple Kanji.
- Define the minimum dot pattern matrix size to ensure sufficient resolution even for the most complex Kanji character.

Language Support for Future Systems

During 1972 through 1974, Masumi Iwao was assigned to several IBM product development laboratories in the US to participate in the development of the IBM 3767 Katakana Communication Terminal.

A designated Kanji project was initiated in 1973 by the Marketing Division of IBM Japan. It was upgraded to a regional IBM Asia/Far East (A/FE) project in the following year, and staffed mainly by personnel from the Fujisawa Product Development Laboratory.

The Marketing Division of IBM Japan also started research projects for Kanji processing jointly with a few customers in 1975. Valuable experience had been gained with the IBM 2245 Kanji Printer system, enhanced by the findings of the Kanji linguistic study.

In the early 1970's, both Toru Takeshita and Toshiaki Igi were assigned to Product Line Management (PLM) at IBM World Trade Corporation Headquarters (WTCHQ, later Americas/Far East or A/FE HQ) primarily to work for the FS (Future Systems) requirements. Both men frequently visited IBM's Data Systems and General Products Divisions (DSD and GPD) development laboratories to explain the hardware and software requirements of Kanji.

Programming Languages

Toru Takeshita, as Advanced Systems Product Manager in A/FE HQ provided preliminary specifications for the inclusion of Kanji in programming languages to projects in IBM's Santa Teresa laboratory.

The Kanji data type was introduced to handle Kanji data and Kanji comments or remarks in coding. Keywords and reserved words in programming languages and file names remained in English. Katakana was no longer contemplated, because the experiment with PL/I keywords in the IBM System/360 era had not found enough use and attention. It was not even considered in the JIS COBOL and PL/I. As subsequent programming languages, for example, Pascal, C, C++, Java, etc. were mainly used for systems and engineering progamming at least initially, very few people were interested in the use of Katakana keywords as it did not make coding significantly faster or easier.

Door-Openers for New Product Developments

In the early 1970s a whole variety of new technologies in hardware and software had become available. This offered opportunities for quite novel product developments, and a full-fledged product development effort commenced in 1977 under the overall coordination of IBM Japan. IBM's entire data processing know-how and technologies throughout the world were to be used. Major new technologies were especially:

- Virtual Memory, a software concept for automatically anticipating information exchange between the limited-size main memory and large backing mass storages. FR Guentsch at Berlin University had first invented the principle in a PhD dissertation, (Ref. 16), but it got lost. Re-invented in the mid 1960s, it entered also IBM's System/370 product line. It was called "Dynamic Relocate" function.
- Data Entry by Keyboard and Display linked under a processing function control.
- High Resolution All Points Addressable (APA) Technology. Applied in displays and printers (Ink Jet, Laser, and Photo Composing), this technology changed the world from the character-based approach to APA which provided the solution for Kanji and Far Eastern Languages representation.

As top priority, the design concept emphasized the need for compatibility of Kanji systems with the existing alphanumeric systems, inclusive of Katakana, instead of isolating any Kanji system as a "dedicated and unique system".

This concept inevitably called for coexistence of new Kanji codes with existing alphanumeric codes without interfering between each other in entirely intermixed usages. As a fundamental requirement, they had to be operable in all Data Base and Data Communications (DB/DC) application software of general-purpose systems.

Another important design objective was to ensure that Japanese customers could perform easy transitions from Katakana data to Kanji by means of software.

To meet these design objectives, development of each Kanji system component, both hardware and software, had to be carried out by those IBM development laboratories that were responsible for the system element. Thus, IBM sites located in various countries across the world became involved:

- Product development of the Kanji control unit, Kanji display unit and Kanji printer that constituted the IBM 3270 Kanji Information System, the core hardware, were carried out by the IBM Japan Product Development Laboratory in Fujisawa, with strong assistance from the IBM Development Lab at Kingston.
- The IBM San Jose Laboratory developed the IBM 3800-2 Kanji printing subsystem.
- DB/DC and high-level language development, which showed the use of the Kanji data type and comments was carried out at the IBM Santa Teresa Laboratory in the US, and the IBM Hursley Lab, UK.
- The IBM 5924-T01 Kanji Keypunch was developed in collaboration between the IBM Japan Product Development Laboratory in Fujisawa and the IBM Toronto Laboratory in Canada. The IBM Raleigh Keyboard Product Engineering Laboratory developed the Kanji Keyboard subassembly.
- The Program Development Center of IBM Japan located in Kawasaki, called the “Tokyo Programming Center“, developed a series of Japanese Language Processing Software programs.
- During all these time periods, the Marketing Division of IBM Japan consistently played the role of leader and negotiator of the projects.

New Kanji System Announced

Upon successful completion of the above mentioned development activities, IBM announced a new Kanji System in September 1979. It consisted of the IBM 3270 Kanji Information Display System, the IBM 5924-T01 Kanji Keypunch, the IBM 3800-2 Kanji Printing Subsystem and associated programming support in form of system software and utility programs.

The system was implemented to function under the control of an IBM System/370 Processor, Virtual Memory Functions of an IBM 303X CPU, and IBM 8100 Information Systems.

The new IBM Kanji System implemented the design concept of compatibility of a Kanji system with existing alphanumeric systems including Katakana, and it was not any more a unique system dedicated to Kanji. The basis of the IBM Kanji System was a well-designed code structure, using two bytes per each Kanji character in an extension to the alphanumeric EBCDIC encoding system.

IBM Japanese Character Set

An IBM Japanese Character set was defined and published in August 1979 for the new IBM Kanji System (IBM manual # N:GC 18-0611, Ref. 24). Its defined 7190 characters became a superset of JIS C 6226 (the Japanese Industry Standard character set). Based on a Kanji usage study conducted by the IBM Japan marketing organization, 388 characters were added to the JIS character set. The IBM Japanese Character Set provided 4370 character code positions for additional characters that a user can define and add for his application needs. An IBM Corporate Standard (C-S 3 3220-024) eventually formalized this IBM Japanese Character set.

IBM 3270 Kanji Information Display System

The IBM 3270 Kanji Information Display System (Figure 38, photographed by Toshiaki Igi), consisted of the IBM 3274 Model 52C Control Unit, the IBM 3278 Model 52 Display Unit equipped with a 12-shift key Kanji keyboard, and the IBM 3283 Model 52 Printer.

The IBM 3274 Model 52C Control Unit contained a UC.5 microprocessor and micro codes for supporting the Kanji functions. The micro codes in the IBM 3274 Model 52 Control Unit interpreted the IBM 3270 Kanji data stream (the IBM 3270 data stream architecture was extended for Kanji data) from the host system and sent it to the IBM 3278 Model 52 Display or the IBM 3283 Model 52 Printer. The micro

code also tracked the shift- and data key depressions on the Kanji keyboard attached to the IBM 3278 Model 52, and generated the two-byte EBCDIC Kanji code for the IBM 3270 data stream.

The IBM 3278 Model 52 Display Unit displayed the Kanji characters in a 16x16 dot matrix, which was generated by Large Scale Integration (LSI) logic and the Kanji font stored in Read Only Memory (ROM), using the two-byte EBCDIC Kanji code. The IBM 3278 Model 52 Display screen could display 960 Kanji (40 Kanji in a row and 24 rows), or 1920 alphanumeric characters in a 7x9 dot matrix (80 characters in a row of 24). Any mixture of Kanji with alphanumeric data was done on a field by field basis.

The Kanji keyboard subsystem was common to the IBM 3278 Model 52 Display and the IBM 5924-T01 Kanji Keypunch. This keyboard differed from the Kanji Keyboard of the IBM 2245 Kanji System, but followed the similar concept of a multi-shift operation. This keyboard featured 12 shift keys and 254 data keys. For direct data entry, the keyboard layout included a mixture of 2,567 alphanumeric, Katakana and most commonly used Kanji characters. In addition, this keyboard provided the input capability for those characters that were not in the layout, by entering the character reference number. The conventional alphanumeric typewriter keyboard layout was integrated into the center of the keyboard for a single-byte alphanumeric data entry operation.

The IBM 3283 Model 52 Printer composed each Kanji character in a matrix of 28x28 ink jet dots. This product was developed by the IBM Fujisawa Development Laboratory, using the electrostatic ink jet technology developed by the IBM Office Products Division in Lexington, KY (later to become the independent Lexmark Company). This technology had a 240 dots-per-inch resolution, producing high Kanji print quality.

In summary, the IBM 3270 Kanji Information Display System facilitated interactive operations between a host system and a display unit that is between IMS and CICS (Information Management System and Customer Information Control System) Kanji applications and a display unit with a Kanji keyboard. The IMS and CICS were enhanced to support the IBM 3270 Kanji data stream.

Figure 38. IBM 3270 Kanji Display System Source: Ref. 13

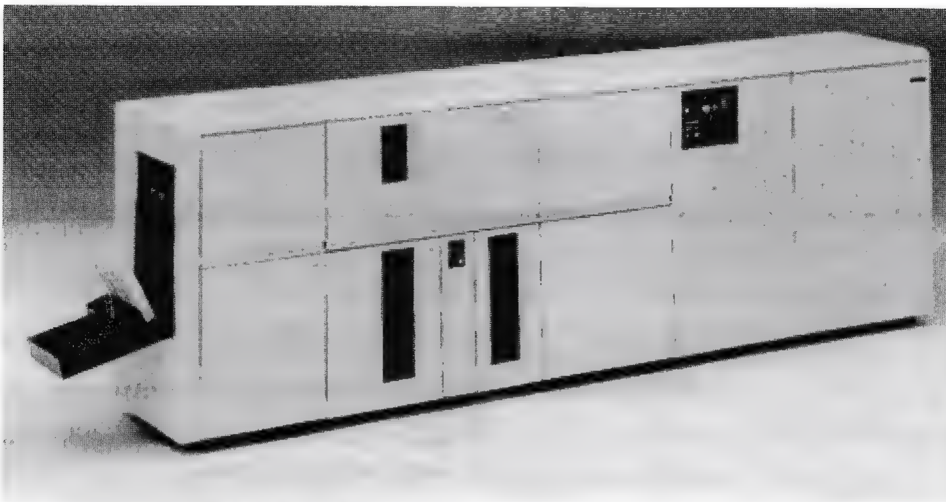
IBM 5924-T01 Kanji Keypunch

The IBM 5924-T01 Kanji Keypunch was based on the IBM 029 and 129 Card Punch units and used the same 12-shift key Kanji keyboard subsystem as the IBM 3278 Model 52 Display unit. The keyboard subsystem was connected to the IBM 029 or 129 Card Punch via LSI logic and generated 2-column punched holes on IBM 80-column cards, using ROM tables for code conversion.

IBM 3800-2 Kanji Printing Subsystem

The IBM 3800-2 Kanji Printing Subsystem was the Kanji version of the IBM 3800 Printing Subsystem, developed at the IBM San Jose, CA Laboratory. It was of huge size, like a trolley car, featuring an ultra high printing speed of 10,000 alphanumeric lines per minute. See Figure 39.

This printer utilized an electro photographic drum, the same technology as in most recent plain paper copiers, continuous paper forms printing.



IBM 3800 印刷サブシステム

Figure 39. IBM 3800 Printing Subsystem Source: Ref. 15

Utility and System Programming Support

Software support expansion for Kanji data type handling was made available by OS/VS, DOS/VSE (Disk Operating System/Virtual Storage Extended) and DPPX (Distributed Procesing Programming Executive) of the IBM 8100 Information System. Kanji data type handling support was provided for Data Base and Data Communication by IMS/VS and CICS/VS. The programming languages PL/I and COBOL were made capable of handling Kanji characters.

Several utility programs were made available, such as one for printing data sets containing Kanji characters, a code conversion program between JIS and IBM Kanji codes, and another conversion program from Katakana data to Kanji.

A very flexible sort/merge program was provided which covered four sorting methods (Basic Kanji type, Japanese dictionary method, and character look-up by both pronunciations, ON and KUN).

IBM Kanji System Follow-on Enhancements

The IBM 3200 Kanji printer was announced in June 1982 as a smaller and more economic printer than the IBM 3800-2. This printer used the same electro photographic drum as the 3800-2 printer.

However, since IBM could not produce a Kanji printer under the timing and price constraints satisfactory to its customers, this was an "Other Equipment Manufacturer" (OEM) product from Hitachi.

The IBM 3273 Model 53 Kanji Printer was announced in October 1981 as another printer of the IBM 3270 Kanji Information Display System. This printer was an impact printer, capable of printing on multiple copy print forms. It employed a 24x24 wire dot matrix print head from the Oki Electric Company.

In July 1980, IBM announced a Japanese Language Processing System (software). Its key elements were the following Kanji application programs:

- a Kanji Document Input/Editing program for the Kanji keyboard;
- a Kanji Document Input/Editing program for the Katakana keyboard with Kana to Kanji conversion capabilities;
- a Kanji Document Composition program;
- a Print Program for the IBM 3800-2 Kanji Printing Subsystem and the IBM 3283 Kanji Printer.

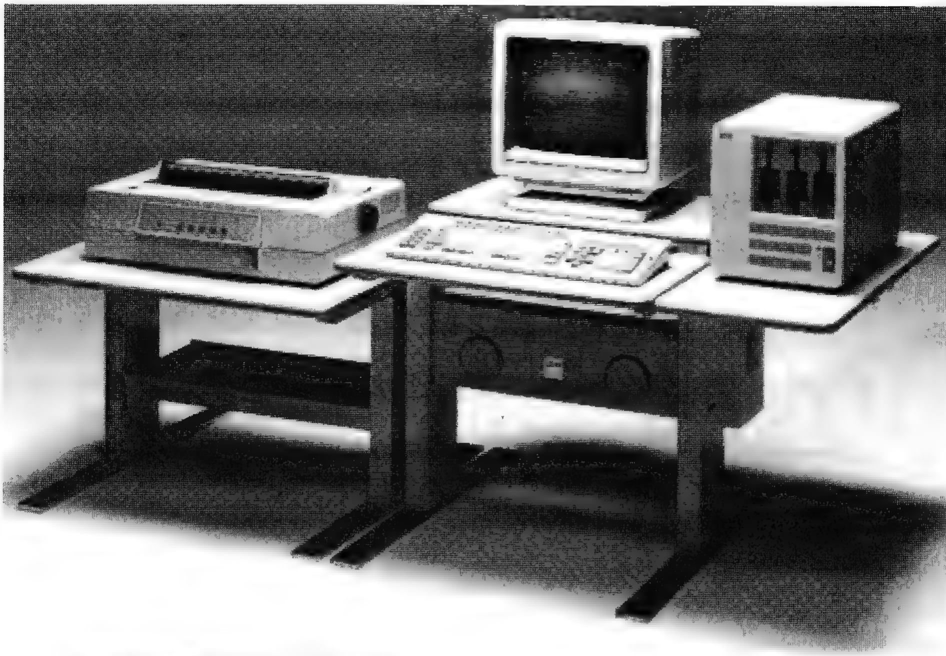
These application programs were supported for MVS (Multiple Virtual Storage system) and VM (Virtual Machine). The development was done by the IBM Japan Programming Center in Kawasaki called "Tokyo Programming Center".

IBM Kanji System/34

An IBM Kanji System/34 was announced in the fourth quarter of 1979. It was developed by the IBM Product Development Laboratory in Rochester, Minnesota. This system provided Kanji interactive display capability similar to the IBM 3270 Kanji Information Display System. The 12-shift Kanji keyboard subassembly for the IBM Kanji System/34 was the same as in the IBM 3270 Kanji Information Display System.

Work Station Business Unit

A Work Station Business Unit (WSBU) was organized in 1982; to develop the Double Byte Character Sets (DBCS) for the IBM PC, to collect customer requirements, make requests to IBM development divisions, and to market workstation products. The Kanji character set for a PC double byte code page called "Shift JIS" was introduced in the Japanese PC industry.



IBM マルチステーション 5550

Figure 40. IBM 5550 Kanji Multistation   Source: Ref. 15

IBM 5550 Kanji Multistation

In 1983, the IBM 5550 Kanji "Multistation" (Figure 40) was announced. It consisted of a desktop Central Processing Unit (CPU) with up to three drives for 5.25-inch 720 KB (kilobytes) floppy disks, a keyboard and a monitor. A wire dot matrix printer came along, providing single style, single size Kanji fonts in a 24x24 dot matrix. The IBM 5550 found wide use as an online workstation in large computer systems, illustrated in Figure 41. However, it was still rather expensive to be marketable to the general public as a stand-alone PC.

The IBM 5550 came also with a Japanese word processor, which was based on the IBM Displaywriter Operating System. Later for the Japanese PC DOS of the IBM 5550, a non-IBM very popular Japanese word processor called "Ichi-Taro" developed by the Justsystem Company was also made available.

Data entry for Kanji was done by a keyboard input front end processor of Kanji PC DOS, which provided Kana (Katakana or Hiragana user-selectable) to Kanji (pronunciation to ideograph) function. The conversion was accomplished by means of a software dictionary and

conversion algorithm on single Kanji character basis, or on Kanji word basis.

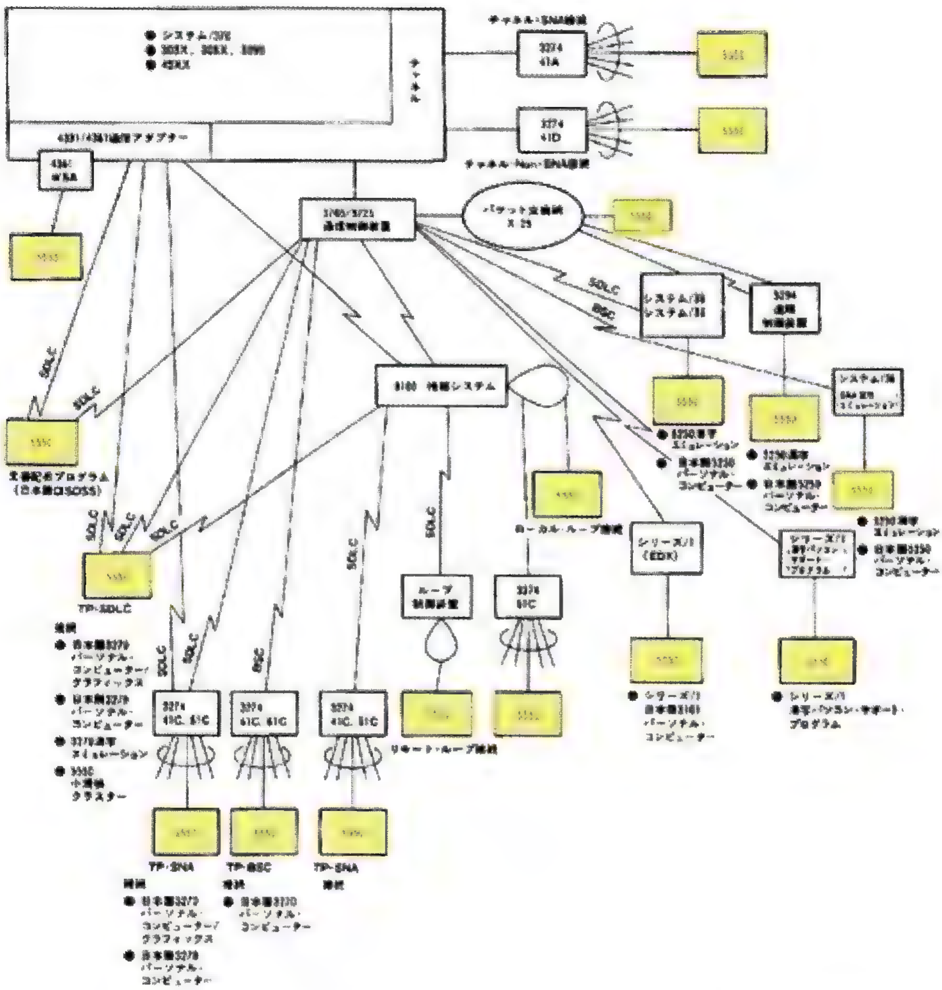


図3-1. IBMマルチステーション・ファミリーとホスト・システムの接続方式

Figure 41 IBM 5550 used as Workstation in a Large System Source: Ref. 15

Double Byte Technical Coordination Office

In order to better support the implementation of the national language requirements in East Asian countries, the Double-byte-character-set Technical Coordination Office (DTCO) was established in 1982. Located at the IBM Japan Product Development Laboratory in Fujisawa, a group of dedicated engineers from IBM Japan was made

responsible for planning, guiding, and coordinating the development of all East Asian language features at all IBM laboratories world wide.

National Language Support

The Double Byte Technical Coordination Office developed an IBM strategy and implementation proposal for National Language Support (NLS) by establishing the technical definition of programming support for DBCS for Japan and Asia. These definitions docoumented the required support based on the application need in each software component. Headquarters of the IBM product development groups (Data Processing Product Group DPPG and Data Processing Communications Group DPCG) were convinced of the need for initiating an organized approach to provide system-wide support for DBCS. It was decided to conduct a corporate-wide task force with major software development laboratory participation. They were to size the modifications of IBM MVS systems software based on the DBCS support level definition.

The Double Byte Technical Coordination Office presented the NLS implementation proposal to IBM Corporation top management in May of 1984. In a favorable response, IBM Corporation set aside a significant amount of money as an approved development fund for five years of work, to modify IBM S/370 MVS software for DBCS support.

Further, in order to ensure automatic NLS implementation in any future product, IBM issued "Corporate Instruction #145, National Language Support", declaring this support to be a fundamental policy.

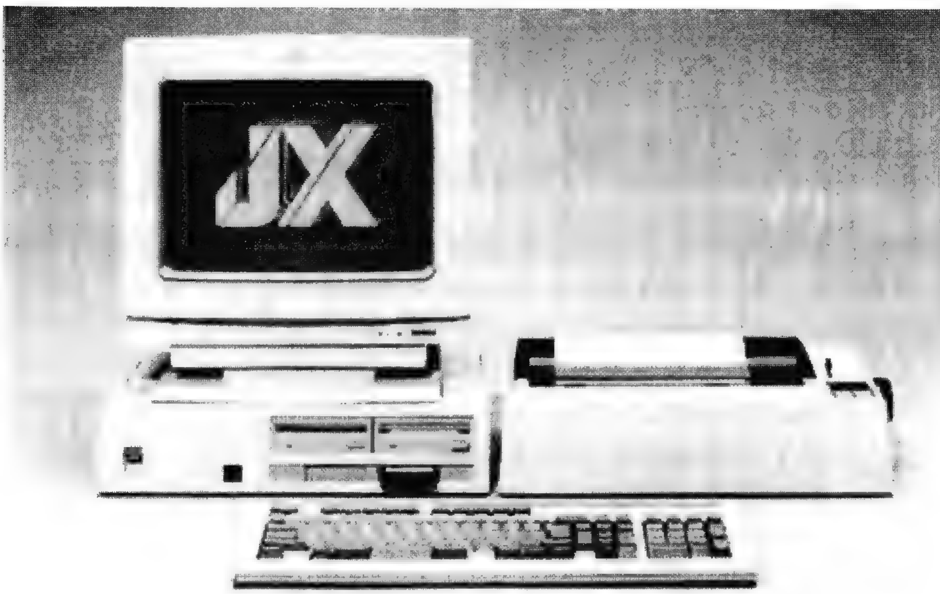
In order to enable global programmers to develop DBCS enabled software products in various IBM programming laboratories world-wide, the DTCO developed and published various English guide books for design, testing and application (Ref. 27, 28). DBCS Competency Centers were organized in five locations at IBM programming laboratories in order to provide on-site technical support, and DTCO sent assignees to these locations.

IBM Personal Computer JX

Not too long after the appearance of the Personal Computer in the United States, IBM Japan followed suit in 1984 with the introduction of its "IBM Personal Computer JX" (Figure 42. Different models came with a tabletop CPU that had one or two 3.5-inch floppy disk drives, or even a hard drive. The IBM Personal Computer JX was one of the first PCs in Japan that introduced 3.5" floppy drives.

The “JX Word” Japanese word processor was using “ATOK”, a phonetic input to Kanji/Kana text conversion program and key part of “Ichi Taro”. A plug-in circuit card housed the fonts.

The customary keyboard cable was the means of basic connection to the system unit. An infrared interaction was a supplemental connection for convenience, offered as an “advantage”. This, however, turned into a handicap if the keyboard was not properly pointed towards the CPU.



IBM パーソナルコンピューター JX

Figure 42 IBM Personal Computer JX Source: Ref. 15

The printer offered a new thermo technology with electrode print wires thinner than those in mechanical devices. The requirement for special thermo-sensitive paper was a drawback. Printing was later replaced by a wire dot matrix. Again, Kanji was available only in single style, single size fonts.

Large advertisement posters from IBM Japan showed a student holding the CPU under his arm, and the keyboard sticking out of his backpack, announcing the first IBM Kanji Personal Computer!

Twenty years after our work on a Kanji solution had started, our dream had come true. It was exciting to buy a JX in Japan and be able to do Japanese Word Processing in any part of the world.

Operating Systems Supporting Kanji

On the software side, the IBM 5550 Multistation and the IBM JX PC ran on a Japanese unique PC DOS called Kanji PC DOS. Although the file system was compatible with English PC DOS, the application compatibility between the two systems was not very good, primarily because of the conflict between Shift JIS encoding and the English encoding scheme.

IBM Japan developed DOS/V, Disk Operating System/VGA (Video Graphics Array), which came as a solution to this problem. This epoch-making contribution eliminated the need for providing a Kanji character font generator in output devices (displays and printers). Competitors (Fujitsu, NEC, Toshiba and others) followed and adopted this method.

Kanji Requirements for IBM Sites Outside Japan

The need for Kanji processing outside Japan emerged first at the IBM Corporate Standards office where I was working since 1977 as Program Director of Standards. One of my responsibilities was the publication of IBM Internal Standards. Earlier, these documents used to be contracted out to graphic designers and publishers for printing. I changed this to a pure internal task, from camera-ready masters to making them electronically available over the IBM internal network.

For the development of future Kanji products, all IBM sites worldwide had to be provided with IBM Standards on coded character sets of Asian languages. Preparations for their publication had to be made.

I designed a layout that would fit on both, the international standard A4 paper size, as well as on the US standard size of 8 1/2 by 11 inches. This was a relatively easy task, but getting documents with good quality Kanji into the hands of all IBM product developers was not.

As a first step, I ordered one of the IBM 5550 Work Stations for my office and used it until my retirement. However, the quality of dot matrix single-style, single- size Kanji fonts was inadequate.

Since the SDD Boulder-developed photo composer had just been installed at a few selected IBM sites, I started using one of them for the

production of IBM Standards documents in-house. However, this method implied that the Corporate Standards office would have to distribute all these documents by mail, which was not acceptable. Print quality was excellent, but the host system had no Kanji fonts. Further, the production of camera-ready masters still required the additional step of chemical processing of the documents at the site of the photo composer.

Here a new IBM 4250 high-resolution printer (Figure 43) offered a less complicated solution. This machine, developed during the 1970s by the Development Laboratory of IBM Germany, could produce brilliant, high quality camera-ready masters, using black paper coated with aluminum powder. Hair-thin electrodes eroded tiny dots and exposed the black paper to compose the letter in a dot matrix so fine that it could be recognized only under a magnification glass. Unlike the photo composer, this technology did not require any additional treatment of the printed output.

I installed one of the IBM 4250 printers in my office and produced a large number of documents for topics other than Kanji. Kanji was still a problem, but a new one dwarfed it, although it may have been an unjustified alarm. When I heard that deposits of aluminum were found in the brains of deceased Alzheimer patients, I stopped using my IBM 4250 immediately and moved to another room. Other more practical reasons for discontinuing the use of this machine were the requirement for the special paper, and the fact that it was not widely available at all IBM facilities. Marketing of this printer was soon terminated anyway when the superior laser printers arrived.

The solution for online retrievability of IBM Standards on Kanji at all IBM sites worldwide eventually came with the IBM 3820 Laser Printer. This machine was installed in every IBM site at multiple locations in large numbers. The lack of Kanji fonts on IBM systems outside Japan was overcome by implementing a unique method with embedded graphics.

The first IBM Standard for a Japanese Kanji character set had to be published in 1979, and the English text was ready. However, there was no Kanji font library on any IBM host system except in Japan. The solution was a page segment (PSEG) feature of the IBM internally used text-processing software called "ISIL" (later announced as an IBM product under the name "BookMaster"). It allowed the insertion of such page segments (graphics) between texts.

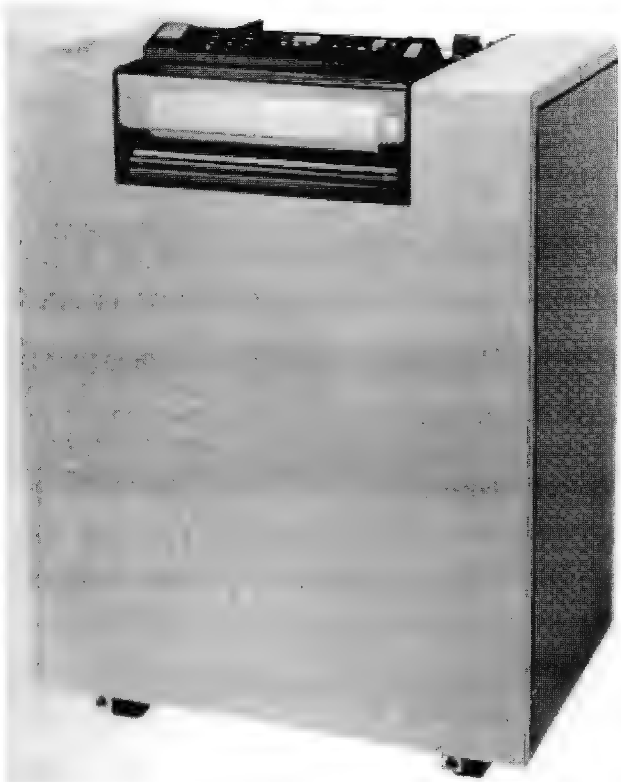


Figure 43. IBM 4250 High Resolution Printer Source: Ref. 15

So, we needed to produce one page segment for each of the thousands of Kanji characters, which appeared to be a monstrous task. However, two brilliant young programmers of IBM Japan, Takashi Ogura and Akio Kido, accomplished it. They devised a programmed semi-manual method to convert the Kanji fonts called "FONT3820" into PSEGs and inserted those into the ISIL source file of the document. It worked, although formatting of the ISIL source file brought the host system almost to its knees.

China

From the mid-sixties to the mid-eighties, a few products with Chinese language capability were marketed by IBM, but only in Taiwan. Specific information proved to be very difficult to obtain, as the present generation of local IBM employees had only a vague recollection of DBCS products marketed in Taiwan before 1984.

Likely products that were involved are the IBM 3283 C52 inkject printer, with the IBM 3278 Display Unit and a large multi-shift data entry keyboard, adopted from the IBM Japan Kanji System. The data entry operator had to remember the positions of all characters precisely. Presumably, there must have been a certain arrangement pattern on the key tops.

As the IBM PS/55 dot matrix could print Japanese Kanji, it is likely that it could have been adopted to print Hanzi.

Likewise, CICS (Customer Information Control Systems) were enabled in order to accept, process and output 3270 DBCS data strings.

Korea

My proposed implementation for Hangul on IBM System/360 was documented in the IBM Special Engineering Catalog of RPQ offers. Therefore, it is conceivable that some products with Korean language capability were actually marketed by IBM between the mid-1960s to the mid-1980s. However, relevant documentation could not be obtained by the time of publication of this book.

Thailand

Supported by Thai RPQ designs of IBM 026/029 Keypunches and IBM 056/059 Verifiers for data entry, a number of products with Thai language capability were marketed by IBM Thailand (see Figure 44). As a Thai National Standard for a coded character set was not published until 1986, the internally defined Thai EBCDIC encoding was used.

Year	Product
1966	IBM System/360 and IBM 1403 Printer
1974	IBM System/3270 and IBM 3277 Terminals, IBM 3287 Printer
1977	IBM System/32
1978	IBM System/34
1981	IBM System/38
1983	IBM System/36

Figure 44: IBM Products with Thai Language Capability Source: Ref. 17

In 1984, IBM Thailand established a "National Language Development" (NLD) team. Its initial task was to develop Thai language support for controllers, printers, display terminals, and specifically required software such as terminal emulation. Later, through the early 1990's, the team assisted IBM development laboratories in implementing the Thai language requirements on IBM products. Over the last few years, the NLD team has been providing not only development coding and testing support, but also educational material, test procedures, translation and localization services.

East Asia Coded Character Set Standardization

Start of Kanji Standardization Efforts

During the development phase of the IBM New Kanji System, which was announced in September 1979, the standardization of Kanji coded character sets became an urgent need. A standing procedure at IBM was that the announcement of any new product required the official sign-off from designated key functions at IBM Corporate Headquarters. One of those was the IBM Corporate Standards function. Consequently, the IBM New Kanji System required the preparation and publication of an IBM Corporate Standard for the IBM Japanese Coded Character Set.

This was only the beginning of a still-ongoing effort, stretching over decades. The IBM Double-byte-character-set Technical Coordination Office (DTCO) was given the responsibility to prepare IBM Standards for the Far Eastern countries. The work had to be based on previously established ground rules for character encoding, available in four fundamental IBM Standards documents:

- IBM Corporate Standard C-S 3-3220-002,
  Extended BCD Interchange Code EBCDIC

- IBM Corporate Standard C-S 3-3220-019,
  Coded Character Sets Implementation

- IBM Corporate Standard C-H 3-3220-050,
  Registry, Graphic Character Sets and Code Pages

- IBM Corporate Standard C-S 3-3220-102,
  Double-Byte Character Set (DBCS) Terminology and Code Scheme

To expect an all-encompassing description of the coded character set standards on Far Eastern languages here would be asking for the impossible. This book can only provide general information with leads to other detailed documentation.

IBM and National Documentation

While the character sets in Western languages are of finite numbers, East Asian languages could be labeled "open-ended". National authorities in those countries define their sets in groups of usage. While most characters find their way into a group that will be encoded for computer applications, there are still many others of lesser usage that

remain "not encoded", perhaps even forever. Revisions are made all the time, which must be reflected in IBM Standards.

Figure 45 shows the list of IBM standards documents, which needed to be published on coding topics for Japan over the last twelve years alone. Each of these documents underwent at least one, but mostly more, revisions during the preceding twelve years listed, in order to accommodate changes. Figure 46 shows IBM Standards documentation activities for China and Korea. Both figures are an indication of the fluidity of the problem.

Japanese Coded Character Sets, IBM

Within the large number of code points of the 16-bit structure (meaning double-bytes), the IBM Corporation defined many code pages on a global scale. In order to be responsive to all national requirements in the future, it was necessary to cover all languages and special characters, plus controls.

Relying on the most comprehensive Japanese dictionary, IBM Japan defined an EBCDIC "Code Page", assigning two 8-bit bytes for each Kanji. At this time, unlike the arbitrary code assignments that were used for the IBM 2245 Kanji System, important logical aspects were considered for the new set.

IBM Standards for Japanese coded character sets emerged in two versions, one for host systems, and the other for PCs. Newly defined JIS (Japanese Industry Standard) character sets were included as "sub-sets". Several subsequent revisions to the IBM Standards became necessary to recognize changes or additions to JIS, and JIS itself came in different versions.

Date	IBM Reference Manuals and Standards	Japanese Industrial Standards (JIS)
1972-04	IBM 2245 Kanji character and code set (7525 characters) Reference Manual N:GA18-1018	
1978-01		JIS C 6226 defined (6,802 characters) (later renumbered as X 0208)
1979-07	IBM C-S 3 3220-024 (7,190 characters) published for Host-code (glyph image based on Seirei *1 and "Shinjigen" *2) (28 symbols and 360 Kanji more than JIS) Reference Manual N:GC18-0611	
1983-03		JIS C 6226 revised (6,877 characters) — 71 symbols and 4 Kanji characters added — 22 pairs of Kanji code point swapped — 244 characters glyph image changed
1986-08	IBM C-H 3-3220-024 revised (7263 characters) — PC code definition published *3 — 69 symbols and 4 Kanji added *4 — 11 characters glyph image changed *5 Reference Manual N:GC18-0785	
1987-03		JIS C 6226 renumbered to JIS X 0208
1990-09		JIS X 0208 revised (6,879 characters) — 2 Kanji added — Heisei font used for standard publishing
1990-10		JIS X 0212 defined, Japanese Supplementary, (6,067 characters)
1992-11	IBM C-H 3-3220-024 revised (7,265 characters) — 2 Kanji added — 14 characters glyph image changed *5	
1993-03	IBM C-H 3-3220-127 published, for Ex tended UNIX Code (EUC)	
1995-01		JIS X 0221, UCS (Japanese version of ISO 10646)
1996-08	IBM C-H 3-3220-133 published for DBCS-PC (New JIS Sequence) added	
1997-01		JIS X 0208 revised, Shift-JIS and Internet encoding added
1999-03	IBM C-H 3-3220-024 revised, Host code (14,823 characters) revised to accommodate Unicode Japanese Subset	

*1 The glyph image is defined by Government order
*2 One of popular Japanese Kanji dictionaries
*3 IBM PC implemented the code earlier as internal process code
*4 Two characters were already in IBM 1979 definition
*5 Adjusted to follow "Seirei" glyph image

Figure 45. DBCS Standards, Japan Source: Ref. 18

Japanese Industrial Standard

There was no Japanese Industry Standard in 1971 when IBM announced the IBM 2245 Kanji printer. IBM filled this void by pioneering the definition and publication of its Kanji Character Set and Code Reference Manual (Ref. 22).

On a national scale for Japan, a JIS standards committee began to develop a standard Kanji coded character set for the country in 1976. Since then, Kanji JIS have been revised and expanded until now. Together with representatives from industry, universities, publishing companies and other experts in this area, qualified and designated IBM employees from the Kanji development team at the Fujisawa Lab, and the Double-byte-character-set Technical Coordination Office (DTCO) continuously and actively participated in this standards committee work of JIS. The goal of this committee was to allocate a unique pair of 8-bit bytes to every Kanji on designated code pages.

Toshiaki Igi was a member for the first JIS Kanji Industry Standard committee, which published JIS C 6226:1978 (subsequently changed to JIS X 0208). Toshiaki's experience from his IBM 2245 Kanji printer character set definition work was of great benefit for the JIS committee. Other key contributions from IBM were made later through Tsuneo Oda, who was the secretary of JIS X 0208:1997, and Akira Oda, who was the group leader of defining the Shift JIS encoding from the de facto standard into a JIS standard in JIS X 0208:1997. These people and unnamed others deserve high praise for their efforts in the development of JIS Standards for Kanji coded character sets. Japanese readers, who are interested in more detail, may wish to consult References 5, 6, 8, 9, 10 and 11.

Development of Industry Standards was tedious and difficult, because the IBM representative on a committee found himself frequently in conflict between the committee objective and IBM's product direction and computer architecture. The task required considerable work and patience. The following examples illustrate the complexities of some of the problems encountered during the years :

- JIS defined the standards for Information Interchange primarily for communication line use, based on the 7 bit / 8 bit ISCII (International Standard Code for Information Interchange) scheme. JIS standards for coded Kanji character sets and the code extension technique of mixing one-byte and double-byte code could not be implemented in the internal code for OS processing in a mainframe, Unix machine and PC. This internal code implementation required a code conversion from the JIS standard by code mapping or table translation.

- In JIS C 6226:1983, the code points for 22 pairs of Kanji characters were swapped, affecting 44 Kanji characters. This created a code

point compatibility problem. IBM could support both JIS C 6226-1978 and JIS C 6226-1983 for Information Interchange since a table translation method was used. IBM kept consistency with the EBCDIC Kanji internal code, and customer data base code integrity. Some other mainframe manufacturers could not support JIS C 6226-1983 since they implemented direct code mapping from JIS 1978 to their internal Kanji EBCDIC code.

- For PC internal code, Japanese Control Programs for Microcomputers (CP/M) and Japanese Microsoft (MS) DOS introduced the Shift JIS method in early 1980. IBM was one of the first in adapting Shift JIS and implemented it for the IBM 5550 in 1983. IBM assigned code page 932 for Kanji Shift JIS in the IBM Standard for Graphic Character Set ID (GCSID). Shift JIS for PC became the de facto standard until it was included in JIS X 0208 in 1997 (Ref. 9).

The DTCO team followed the JIS activities very closely, to ensure that future IBM products would support all of the nationally standardized Kanji. In order to continuously meet this objective throughout the following years, they produced and maintained an internal IBM document titled "IBM Coded Character Set for Japan" (Ref. 23, 24, 25). Derived from IBM's Kanji usage study for data processing, this document always included additional Kanji characters over and above the JIS Kanji set. IBM declared that compatibility would be maintained with JIS, by always making the IBM character set a "superset" of JIS when JIS changed. Figure 45 is the history overview.

Eventually as a result of these activities over decades, an official JIS Kanji Dictionary emerged (Ref. 5). It lists all Kanji with their pronunciations, along with their double byte code assignments in all relevant code schemes. A designated special section provides the means with which any Kanji can be found by its basic stroke pattern. This book is a masterpiece example for years of hard, tedious, and detailed work.

Chinese Coded Character Sets

For China, the DTCO prepared two IBM Standards for Chinese coded character sets. Assisted by native employees from local branches in Taiwan and Hong Kong, one document addressed "Traditional Chinese" (center column of Figure 46). A second effort, undertaken with native employees from Mainland China resulted in an IBM Standard for "Simplified Chinese" (rightmost column of Figure 46. Every Chinese Hanzi was allocated to a unique pair of 8-bit bytes on designated code

pages. Subsequent revisions to the Chinese standards were required, due to changes that were made by national standards committees.

According to Lunde (his book "CJKV", Ref. 2), the Kanji characters used in Hong Kong and Viet Nam are based on Traditional Chinese, but they have their own unique standards.

For China, the DTCO prepared two IBM Standards for Chinese coded character sets. Assisted by native employees from local branches in Taiwan and Hong Kong, one document addressed "Traditional Chinese" (center column of Figure 46). A second effort, undertaken with native employees from Mainland China resulted in an IBM Standard for "Simplified Chinese" (rightmost column of Figure 46. Every Chinese Hanzi was allocated to a unique pair of 8-bit bytes on designated code pages. Subsequent revisions to the Chinese standards were required, due to changes that were made by national standards committees.

According to Lunde (his book "CJKV", Ref. 2), the Kanji characters used in Hong Kong and Viet Nam are based on Traditional Chinese, but they have their own unique standards.

	Korean	**Traditional Chinese**	**Simplified Chinese**
IBM Standard	1985-03 Defined: Korean DBCS-Host code	1985-01 Defined: Traditional-Chinese DBCS-Host code	1985-01 Defined: Simplified-Chinese DBCS-Host code
	1989-10 C-H 3-3220-124 published DBCS-Host and DBCS-PC	1989-10 C-H 3-3220-126 published DBCS-Host and DBCS-PC	1992-11 C-H 3-3220-130 published DBCS-Host and DBCS-PC
	1992-09 C-H 3-3220-125 published DBCS-Host and DBCS-PC revised	1992-01 C-H 3-3220-126 revised DBCS-Host and DBCS-PC	1993-11 C-H 3-3220-130 revised DBCS-Host and DBCS-PC
	1993-11 C-H 3-3220-128 published Extended UNIX Code (EUC)	1993-11 C-H 3-3220-129 published Extended UNIX Code (EUC)	1994-06 C-H 3-3220-132 published Extended UNIX Code (EUC)
	1997-09 C-H 3-3220-030 revised DBCS-PC, industry standard code	1994-01 C-H 3-3220-131 published BIG 5 Code (industry standard PC)	1996-02 C-H 3-3220-020 published for GBK Code
	1999-04 C-H 3-3220-030 revised Euro symbol added	1999-04 C-H 3-3220-126 revised Euro symbol added	
National	KS X 1001:1992, KS X 1002:1991, KS X 1003:1993; KPS 9566-97 (North Korea)	Big Five, Big Five Plus, CNS 11643-1992, CCCII (75,684 characters)	GB-Series (11 documents) see Lunde's book (Ref. 2) page 72
Note: The bottom row provides only a lead to Korean and Chinese national standards documents. For any detailed information, consult with Ken Lunde's book (Ref. 2), and also the books on "Unicode" (Ref. 3, 4). Japanese readers can obtain more details			

Figure 46. DBCS Standards, Korea and China Source: Ref. 18

Korean Coded Character Sets

The standardization of Korean character sets was comparably complex. The leftmost column of Figure 46 summarizes IBM and national documentation. The Kanji characters used in Korea are based on Traditional Chinese. North Korea has its own standard, which differs from the South Korean document.

Each of the Jamo, Hanja, and the large number of Hangul permutations were given a unique, two 8-bit byte assignment on designated code pages.

Thailand Coded Character Sets

After IBM Thailand had established the National Language Development (NLD) function in 1984, its representatives began working with the Thai Industrial Standards Institute (TISI) and other vendors to define national standards. This work resulted in the issuance of TIS 620-1986 on the Thai Coded Character Set Standard. A revision was published in 1990.

IBM internal standards for Thai were developed in compliance with national and international standards. They are registered as code pages number 874 (PC-ASCII) and 838 (EBCDIC), as part of IBM Standard C-H 3-3220-050, "Registry Graphic Character Sets and Code Pages".

Unlike Korean, IBM Thailand decided to only standardize code pages for the set of components in a single 8-bit byte structure, but not the symbols themselves. IBM Thailand wanted to stay in single-byte, because the DBCS PC at that time required special hardware like display adapter cards and fonts. They developed a small rendering engine (a print program routine) to display composed symbols from variable byte encodings, which simulated a single-byte data stream from the view point of application programs. This enabled them to use many English application programs without modification. Business size was relatively small in Thailand, and the first requirement was to be able to handle Thai data. But the special efforts that would be required to translate all of the programs weere to be avoided.

International Code Page System

National character set standards for all languages had to be prepared in concert with an international effort conducted by the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC) in a joint technical committee ISO/IEC JTC1/SC2 for a global structure of "Code Pages" documented in ISO/IEC

10646. It is the key standard on Far Eastern Languages, "ISO 10646: Information Technology – Universal Multiple-Octet Coded Character Set", but was not published until 1993.

IBM Japan code experts closely coordinated the Kanji coding with a designated function of IBM Corporation, the "Globalization" department, which had worldwide responsibility for the project of "Coded Character Sets Standardization". Careful consideration was essential, so that national code page assignments would not conflict with each other. Thus, only certain areas of code points were selected for Kanji. This was the beginning of an effort that went on for decades on an international scale and is still ongoing.

ISO/IEC 10646 is a fundamental standard, potentially affecting all parts of information technology. However, it specifies only coded character sets, not a complete system for text representation. It provides the basis for internationalization, but does not give a perfect solution to the problems in this field.

Outlook Universal Code

First published in 1993, the Universal Multiple-Octet Coded Character Set System, in short "Unicode", is a character set specified by a consortium of major American computer manufacturers. Its primary objective was to overcome the chaos of different character sets in use with multilingual programs, creating software for international applications.

Unicode assigns a unique number to each character in each of the major languages of the world, in order to eliminate any conflicts or incompatibilities during information interchange. It started on a 16-bit basis (UCS-2), but a later version was based on 31 bits (UCS-4). With a clever system of mapping onto different 16-bit planes, it has the potential to cope with over one million characters.

Note: The international "Code Page" system is too complex to be elaborated further in this book. For detailed information, consult Ken Lunde's book "CJKV" (Ref. 2). Other recommended sources for information are References 3 and 4, the latter a similarly voluminous book, over 1200 pages entitled "The Unicode Standard, Version 3.0", published in April 2000. In addition, there is a Unicode web page: http://unicode.org.

Major Changes through New Technologies

Emergence of a Broad New Industry

Several major breakthroughs occurred during the 1980's, which threw the gates open for opportunities in the area of computer related technologies. Many companies took advantage of this, with IBM as one of the major players in the field. Therefore, the scope of this book goes beyond the IBM history to the extent necessary, in order to realistically describe developments and events.

A whole new industry emerged, competing with innovative hardware products and related software support. New designs for PCs and peripheral equipment started to become available at short intervals. With these, development of Kanji systems of high quality and affordable to small users became a reality. Millions of people in China, Japan, Korea, and Thailand were finally able to utilize Internet and e-mail as everybody in alphabet-based countries. The following paragraphs describe the enabling factors and accomplished practical methods.

Main Memory

Integrated circuits on chips had long replaced magnetic core memory. Random Access Memory (RAM) capacity grew larger and larger, while processor speed became faster and faster. Interestingly, PCs were selling for less and less, while offering more and more power.

Disk Storage

Progress in storage technologies literally exploded. It started with the 720K low-density floppy disks, soon to be doubled and quadrupled. Hard disk drives of "high capacity" bought today were obsolete tomorrow. Hard disks have become smaller, cheaper, faster and of greater capacity than anyone could have imagined ten years earlier. Other yet more powerful media, like compact disks (CDs), and digital video disks were then easily predictable.

Printing Technologies

Laser technology, fast and noiseless, approached a stage of perfection, which facilitated building machines that outclassed all previous impact printers. The IBM 3820 Printer was one of the early, successful laser printer products of IBM.

In addition to laser, the technology of ink jet printing came along.

Both based on the dot pattern principle, these technologies allowed printing at much finer resolutions than mechanical wire dot printers could ever offer. High quality printing became possible, with a large variety of fonts and at low costs. Ink jet printers especially turned out to be attractive for PC users with rapidly dropping prices. These breath-taking evolutions provoked an abrupt end to impact printing in the mid 1980s.

Multi-Font Kanji Printers

Since converting its Lexington Office Products facility into the separate company "Lexmark", IBM stopped making small printers under contract for several years. Therefore, users of IBM PCs were forced to buy printers from other manufacturers, one of them Lexmark. Others are Hewlett-Packard, Epson, Canon, etc. As these are either laser or ink jet, printing fine dot pattern matrices, they can all handle Kanji, and all other Far Eastern letters, without any special features or modifications.

Dictionaries and font sets usually reside on hard disks, together with programming routines that handle pronunciation group displays.

A large selection of type fonts emerged for practically every written language. It is not enough to accommodate thousands of characters, but there is a need for a multitude of different font styles and sizes for each such character.

Computerization of Kanji Fonts

By exploiting significant advances in high density, low-cost memory technology, the use of high quality and multiple Kanji fonts became the way of life even for PCs. High resolution ink jet printers attached to PCs can now produce almost the same quality Kanji printing as was possible only by expensive Kanji publishing systems like JPS. Kanji fonts evolved from dot matrix representations of only 16x16 or 24x24 in bitmap format to a range from 8x8 to 96x96 in both bitmap and outline format.

Although the Kanji font is in common with the alphanumeric font format, many Kanji character shapes are complex, for which certain special techniques have been implemented:

- Bitmap Font: For the small Liquid Crystal Display (LCD) size of cellular telephones and Personal Digital Assistants PDAs, bitmap fonts of 8x8 or 12x12 are used for Kanji. As this matrix size is not sufficient, the font design takes advantage of human optical illusions for Kanji recognition.

- Outline Font (True Type, Adobe Character Identifier or CID, Open Type): Outline font is a vector representation of the outline shape of the character. One outline font can be used for generating multiple matrix sizes by rasterizing to various matrix sizes. For Kanji, the following technique was introduced:

    1. In case the target matrix size is below 32x32, the rasterized result does not produce a legible Kanji. Therefore, a hand-tuned bitmap font is used, which is imbedded as part of the outline font.

    2. Elimination of jagged contours of the generated bitmap font by grading the outline of the character shape.

Since the Kanji character set is infinite, a font design editor called "Gaiji (outside character) Editor" to create unsupported Kanji fonts is mandatory. This capability had to be provided from the advent of Kanji implementation in computing, and was part of the IBM 2245 Kanji Printer System. A Gaiji Editor is also provided with the Windows Operating System.

High-Speed Impact Printer

IBM announced early in the year 2002 the IBM 5400–L10, a high-speed impact printer with multiple fonts and 24x24 and 26x26 dot matrices. The maximum speed is 1,000 lines per minute. It can print up to 7 copies in addition to the original, and has some other good features. Website address (in Japanese):
http://www.ibm.com/jp/printer/hardware/5400_l10.html

Identical Monitors for Latin and Far Eastern Characters

Monitors of the early years had their own code generators for Latin character sets, and text-processing applications mushroomed. However, for all Far Eastern symbols, headed by Kanji and in particular for small PC users, all-points-addressable (APA) monitors were needed as well as exceedingly fast access storage to grab letters from font libraries and dictionaries. Both came!

Good-Bye, IBM Punched Cards

Lower cost monitors connected to processors resulted in the obsolescence of IBM punched cards. Direct data encoding with input into CPU's and memories via computer terminals called "work stations" became reality. Hard-wired and dial-up connections via acoustic couplers and modems emerged, the beginning of pre-Internet Local and Wide Area Networks (LAN and WAN).

Compact Disks, Memory Sticks, and Smart Cards

While punched cards are surely a thing of the past, CDs might soon suffer the same fate with the appearance of high capacity memory sticks and smart cards, alternately usable with PCs and digital cameras. Attachable to newer PCs via so-called USB connections, they might become a superior competitor for CD drives in the not too distant future.

Data Entry Method for Japanese

With punched cards no longer in the picture, data entry methods settled on the use of keyboard layouts that were very similar to the physical arrangement of keys on English keyboards. In order to permit the entry of Japanese text, however, the key tops of workstations, desktop PCs and laptop computers carried Hiragana symbols in addition to the familiar English QWERTY arrangement. An example of such a keyboard is shown in Figure 47, of the IBM laptop (ThinkPad) computer G40. Users can choose to enter Japanese text by pronunciation in either Hiragana or English alphabet mode. Interestingly, however, even many Japanese people prefer the Latin for selection.

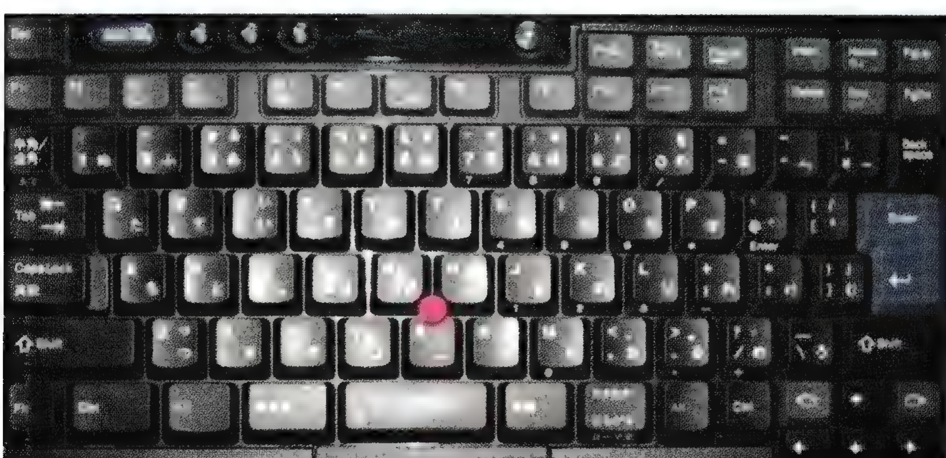


Figure 47. IBM G40 ThinkPad Japanese Keyboard

In addition to the key tops with graphics, the described keyboards have several additional control keys not seen on western keyboards. For example, a special key switches full size characters (double byte) to half size (single byte) and back. Latin characters, numbers, specials, and Katakana are encoded in single and double byte and require this feature.

These enhanced, versatile keyboards need a powerful software support to fully handle double-byte coded character languages. The Justsystem Company was leading this development with its program called "ATOK". It uses not only a pure Japanese dictionary (similar to the English Thesaurus), but also certain syntax and grammar rules to find the most suitable Kanji first. Microsoft implemented this method in its software called "MS IME".

As text is entered by pronunciation, a string of Hiragana appears on the screen. Touching the space bar activates the program to search for and display Kanji where available. Sometimes, several Kanji are displayed to select from. Pointing to the desired Kanji followed by pressing the Enter key moves the character into the text stream of the data file to be created. The program scans through the entire Japanese dictionary (not only a list of individual Kanji characters). It converts Hiragana into Kanji where appropriate according to Japanese language grammar and syntax rules.

When ambiguities occur, the process stops and prompts the user to choose the desired Kanji from a displayed group. Latest software has reached a sophistication whereby entered text could result in several Kanji, whole phrases or sometimes complete full sentences. Once confirmed, the program tags every character with its designated double-byte code for further processing, such as display, printing, and storing.

The major breakthroughs in the area of data entry became possible only through the availability of low-cost storage, such as hard disks, combined with fast processing speed. Whole dictionaries can be stored with the advantage of keying in the pronunciation of words that consist of more than one Kanji character and added Hiragana as needed for grammatical reasons.

Thus, the selection of Kanji by pronunciation on a single character basis came to an end. As a consequence, other previously ventured approaches were abandoned, such as selecting a Kanji character from a huge template, either by a stylus or by multi-shift keyboards.

Data Entry Method for Traditional Chinese

Chinese data entry is a multi-step process. Unlike Japanese with its ambiguities of several pronunciations for the same Kanji, Chinese Hanzi basically have only one, which typically remains unchanged even when combined with another Hanzi. Compared to Japanese, this seems like an advantage, but a tone (flat, up, down and up, or down) superimposed on the pronunciation may change the meaning and, therefore result in a different Hanzi.

Several keyboard layouts and systems emerged, all based on the English QWERTY physical arrangement. Some use the pronunciation and tone approach, while others rely on stroke-pattern input. Individual keyboards with corresponding key top arrangements are available.

IBM came out with a universal combination of the QWERTY layout, featuring the Latin characters together with three additional Chinese symbols on each key top (Figure 48). The Chinese version of Microsoft Windows includes appropriate software support, which allows the user to toggle between several keyboard layouts to activate the desired one. The software package also permits a display of the selected keyboard on the screen, called "soft keyboard", which facilitates data entry without a physical keyboard.

Figure 48. IBM Keyboard for Traditional Chinese Source: Ref. 19

The two Chinese characters at the bottom of each key in Figure 48 reflect two of these layouts with names like "Cangjie" and "DaYi". Under

complicated procedures, a Hanzi can be identified by its basic radicals and moved into the text stream. In addition to the Chinese fonts sets, each such layout must be supported by a designated software package, because the character selection methods differ.

The upper right position on each key reflects the "Zhuyin" system layout, the use of which will be explained in more detail. In this system, a set of 37 sound symbols plus 4 tone identifiers has been devised. The symbols are not radicals. The data entry operator must select one out of 379 defined symbol combinations by sequentially pressing one, two, or three of the 37 symbol keys to identify the pronunciation of the desired Hanzi. Native Traditional Chinese users know, and can use these symbols directly. Other users, perhaps many, and even native Chinese people may look up the alphabetic pronunciation from a list.

Zhuyin	Hanyu Pinyin	Tongyong Pinyin
注音	漢語拼音	通用拼音
ㄔㄨㄢ	chuan	chuan
ㄔㄨㄤ	chuang	chuang
ㄔㄨㄟ	chui	chuei
ㄔㄨㄣ	chun	chun
ㄔㄨㄛ	chuo	chuo
ㄘ	ci	cih
ㄘㄨㄥ	cong	cong
ㄘㄡ	cou	cou
ㄘㄨ	cu	cu
ㄘㄨㄢ	cuan	cuan
ㄘㄨㄟ	cui	cuei
ㄘㄨㄣ	cun	cun

This Figure shows a few lines extracted from the list of the 379 valid symbol combinations sorted alphabetically. The left column of the list shows the symbol combinations, requiring between 1 and 3 keystrokes. Taiwanese authorities are discussing possible alphabetization of pronunciations, trying to decide between the center and right column. The center column headed "Hanyu Pinyin" is identical to the alphabetized Simplified Chinese, which has been in use on Mainland China for over forty years. The shaded lines indicate that there are differences in some cases.

Following the entry of the Zhuyin pronunciation symbols, the tone mark entry is required. It takes into account the melodic Chinese pronunciation for the desired Hanzi. One of the four levels shown in Figure 49 must be selected by pressing the corresponding key:

After this last entry, a list of Hanzi meeting the entered criteria is displayed, as captured in the left box of Figure 50. From those, clicking on it or pressing the corresponding number key can obtain the desired Hanzi. Thus, the entire process of getting one Hanzi on the screen will

take from 2 to 5 keystrokes. Once the desired Hanzi is found, it is tagged with its designated double-byte code for further processing, display, printing, and storing. A further step of sophistication is in the software that sometimes displays a second Hanzi for a likely combination.

Tone Type	Key to be Pressed
flat tone	space bar
raised tone	number 6 key
tone lowered and raised	number 3 key
tone lowered	number 4 key

Figure 49. Zhuyin Tone Selection Criteria Source: Ref. 19

Data Entry Method for Simplified Chinese

Simplified Chinese data entry itself has been greatly simplified, by using only the regular English QWERTY keyboard. The "Pinyin" list is used directly, but may require as many as 6 keystrokes. A tone mark selection is not used, however, which results in a larger number of displayed Hanzi candidates (Figure 50 right box). The upper right buttons permit scrolling through additional selections.

Comparison, Traditional versus Simplified Chinese

Both boxes of Figure 50 show the Hanzi 請, which has the meaning of "Please", the first character of the Chinese sentence in Figure 4.

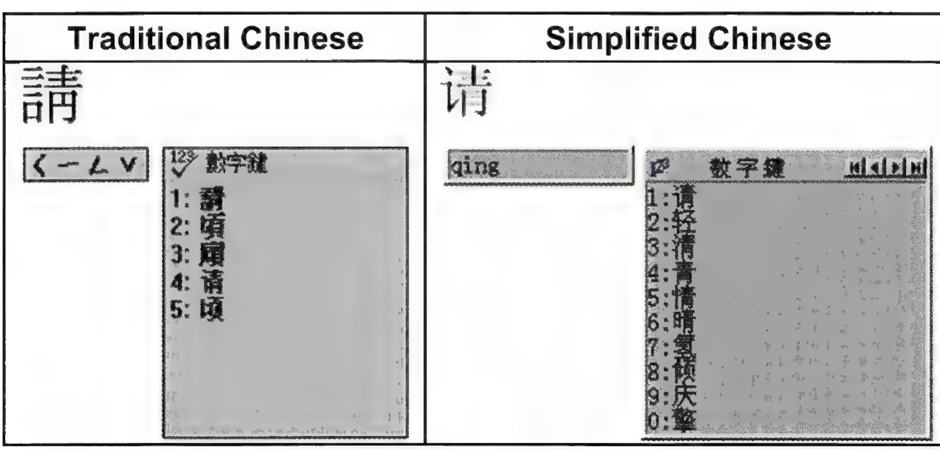


Figure 50. Hanzi Data Entry Screen Displays Source: Ref. 19

Below this Hanzi in the left box, the three designated Traditional Chinese Zhuyin pronunciation symbols of the initially pressed keys are shown, followed by the tone mark. The right box of Figure 50 shows the

corresponding simplified Hanzi, with the previously keyed in pronunciation using the 4 alphabetical letters "qinq". The large shaded boxes illustrate the number of Hanzi candidates narrowed down to 5 for Traditional Chinese, reflecting the tone mark entry. However, many more Hanzi must be inspected in the case of Simplified Chinese.

Science Fiction in Sight

Very recent research raises hope for an increase in efficiency for Chinese data entry. The approach is to have the computer know where its user is looking, or "gazing". A paper in the *Communications of the ACM* (Ref. 12) reports an experimental eye-tracking system that could be used for selecting a Hanzi character simply by gazing at it long enough amongst a displayed group, and then pressing the space bar. This would eliminate the presently needed last step of character selection, whereby the user must take the eyes away from the screen in order to press the designated number key.

Data Entry Method for Korean

The Korean key tops in Figure 51 show the basic phonetic symbols called Jamo, which are consonants and vowels. The Jamo set is a bit larger than the available key tops, forcing some Jamo symbols onto the upper shift location in the QWERTY row. When entering two to four Jamo symbols from the keyboard, those are combined under certain rules and arranged into one character print or display position. Once found, the Hangul character is tagged with its designated double-byte code for further processing, display, printing, and storage.

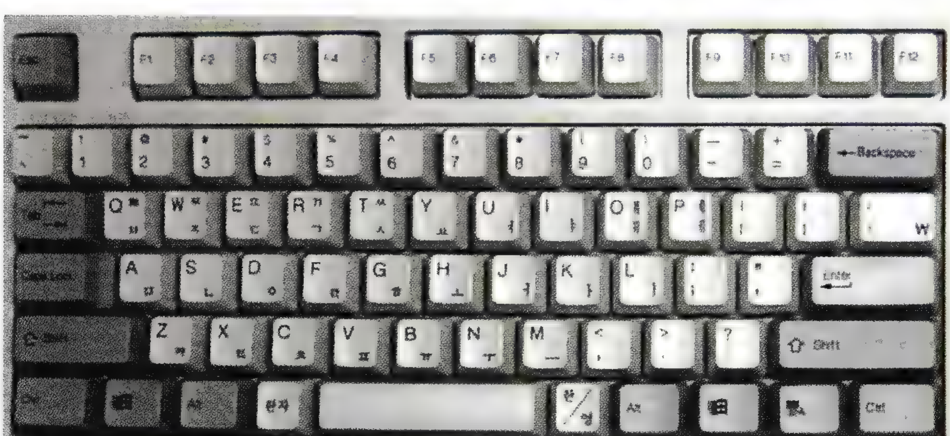


Figure 51. IBM Korean Keyboard Source: Ref. 19

Data Entry Method for Thai

Unlike the Japanese, Chinese, and Korean keyboards, the key tops on theThai keyboard carry no English alphabet, but only the 87 Thai symbol components. These are the consonants, vowels and tone marks that are used to compose a complete Thai symbol. Numbers are shown in Thai, and only some are placed on the same key tops as on the QWERTY layout (see Figure 52). Also unlike Japanese, Chinese, and Korean, the composed Thai symbols (over 2000) are not double-byte encoded and, therefore, the inherent programming support as well as translation work is not required. The penalty for this advantage, however, is that English must be used with many application programs.

Automatic (program controlled) composing of Thai characters was introduced during 1987-1989.

IBM Thai keyboards are registered under the numbers 190, 191, and 192 in the IBM Keyboards Standard document.

Figure 52. Thai Keyboard Source: Ref. 17

For data entry of mixed Thai and English, the user must switch between the Thai keyboard and the English keyboard. Most operating systems allow such language switching by a certain key combination. Arabic numerals are available on the English layout.

Data Entry on Personal Digital Assistants

Originally starting out as wireless cellular telephones, these tiny pocket-sized PDAs soon turned into something much more powerful. The owner of such a gadget not only obtains his personal telephone number, but also an Internet address. E-mail messages can be exchanged with other PDAs or between PDAs and computers. Newer models are equipped with a small video camera, allowing taking pictures and attaching them to the e-mail.

These hand-held devices from many manufacturers accommodate a multitude of Kanji characters, thanks to the breakthrough and progress in micro memory circuit chips. Data entry is phonetic, but cumbersome. The same numeric button must be pushed repeatedly to reach a desired letter, usually with the thumb of the hand that holds the telephone. The PDAs are extremely popular, nevertheless.

A considerable improvement may be a PDA that projects a “soft” keyboard on the table (above picture). Observers at a recent computer exhibition in Asia were amazed to see the Samsung PDA display show the letters, as they were laser-beam traced while being “typed” on the imaginary projected keyboard.

Although the picture shows only an English QWERTY keyboard layout, one could assume that Korean and Japanese might follow.

Operating Systems

OS/2 was used initially, but today, Microsoft Windows runs on most IBM and compatible (clone) PCs. Microsoft Windows is now the most widely used operating system in Japan and the entire world. Microsoft did a good job with national versions of their product by translating inscriptions on icons, tool bars, pull-down menus, help functions, error

messages, etc. It is not possible, however, to translate 100%, and various English terms might always remain unchanged.

Text Processors

For Japanese text processing, IBM products can make use of a variety of non-IBM software products, such as Ichi-Taro and EGWord, both local products purely in Japanese. Microsoft Word comes as part of MS Office, with its other components Excel, Access, PowerPoint, all with Japanese language support.

Considerable effort went into making Japanese word processing possible on US (English) based PCs. This was to serve Japanese users in the United States, in order to avoid the need for buying a second (Japanese version) PC. One such product was a “Kanji Kit”. It required that the hard drive be partitioned, thus simulating two computers in one PC.

A second software product called “KCOM3” from the Kureo Company in Vancouver BC runs on an English version PC. It does not require the hard drive to be partitioned. However, in Japanese mode, many functions compared to a pure Japanese word processor are missing. For non-Japanese users, this software is most user-friendly with all information on task bars, pull-down menus, etc. in English. A click on “Mode” can switch the information back and forth between English and Japanese. However, the functions are only reduced rudimentary subsets of the full Microsoft Word software.

For Microsoft Windows 95 and up, users could download a feature from MS Word 2000 called “Global IME”. Although better than the others, running this feature on an English language PC is not an equal substitute for a Japanese version PC.

Voice Entry of Japanese Words and Text

An interesting product appeared recently that makes use of the available coded and digitized Kanji in storage. Soon after the English version of “IBM ViaVoice” was introduced, IBM Japan developed the Japanese version. This software receives input in the form of spoken words and sentences, finds the corresponding digital voiceprints in a stored Japanese dictionary and displays the associated characters. After a training session to capture and store the acoustic characteristics of the user’s individual voice, an accuracy of over 90% can be achieved. Although subsequent editing is needed in many cases, the main task of data entry via keyboard disappears.

In April 2002, IBM Japan announced "ViaVoice with ATOK15" which integrates ViaVoice Version 9 with the latest version of the Kana-to-Kanji conversion program developed by the Justsystem Company. This allows voice entry of a text anywhere pointed to by the cursor in an existing text, table, etc. created by most Windows software.

WebSphere Voice Server for Transcription

This product is a specialized, server-based, large vocabulary speech recognition package aimed at solution developers and service providers. Based on IBM ViaVoice technology, WebSphere Voice Server for Transcription provides large vocabulary-deferred recognition functions that can be integrated into document workflow applications. This technology supports the conversion of audio into text from a variety of input devices, based on the user's personalized voice model.

The Transcription Server is packaged with a runtime server, Application Programming Interfaces (API), programming and system administration documents, utilities, and sample client code. The Transcription Server functions include:

- transcribing audio to text;
- creating/managing speech user profiles;
- personalizing speech profiles;
- basic server management;
- results log reporting.

As it combines Java and VoiceXML (eXtensible Markup Language), Voice entered from fixed and mobile phones is converted to Japanese text and transmitted to a web application server for such systems as CRM (Customer Relationship Management) utilizing CTI (Computer Telephony Integration). Website address (in Japanese):
http://www-6.ibm.com/jp/voiceland/enterprise/websphere.html

Bilingual Translation

IBM Japan recently introduced a most amazing software package called "IBM King of Translation". It translates Japanese to English and vice versa, either from typed-in text or received as e-mail. Output is not only in a second data stream, but also as voice, expanding on the ViaVoice and type font software packages.

Optical Character Recognition (OCR)

I have been using OCR to scan English and German, and the accuracy is amazing. This is just another example of an application that relies on the fast accessibility of a stored dictionary with digitized letters and fonts.

Recalling the time when I worked at the IBM Development Laboratory in Germany over forty years ago, OCR was a major project that was on the verge of being abandoned. Processor speed and storage space were simply inadequate to accommodate and manipulate vast amounts of data fast enough. A few decades later, there is an abundance of all!

IBM Japan is currently marketing the "IBM Japanese OCR SoftReco" for Windows. It can recognize numerical data at the rate of 1,000, and Kanji data at the rate of 75 characters per second. Fax devices or TWAIN (Technology Without Any Interested Name) scanners are used for input. Website address (in Japanese):
http://www-ibm.com/jp/software/softreco/

Internet

Microsoft Outlook Express, Netscape, Eudora, and others all came up with Japanese versions of their software. I did not specifically check this out, but Chinese and Korean are probably available as well for Internet users in those countries.

IBM Japan has built a "Home Page Builder", which has been selling very well and has won high reputation in Japan. It is included in the best-selling web application server product called "Websphere". Website address (in Japanese): http://www-ibm.com/jp/software/internet/hpb/

It is now permissible to use Kanji as part of an IP address and URL. Japanese character Internet Domains can be registered in Hiragana, Katakana and/or Kanji. Website address:
http://www.japanregistry.com/faqs.php/

Epilogue

Recognition

A standing rule of the IBM Corporation was to recognize key individuals who performed the original work that culminates in the successful announcement of a new product. Thus, in the fall of 1971 "IBM Outstanding Contribution Awards" were given to Massi Iwao and me for our ground-breaking work to realize the first IBM Kanji System.

Continuation of Developments

However, this was only the beginning of four decades of developments, which have brought a multitude of solutions for computing using Far Eastern languages. It required an intense effort on my part during the last year prior to completing this book. It turned out to be a full-time job with frequent visits to the IBM Japan Lab at Yamato to consult with various experts. I found confirmed that developments are by no means coming to a halt now. I have to draw a line here and will leave it up to the young generation when their time comes to document the next forty years of History of Far Eastern Languages in Computing.

Some Problem Areas Requiring Solutions

Considerable work is still required to remove the unfriendliness for the users in multilingual Internet applications. There are problems remaining and future challenges. In the following paragraphs, I am highlighting some aspects that in my opinion require attention for improvements. It will probably take five to ten years to develop solutions, as long as there are PC users who are working with back-level equipment and software.

Many aspects must be considered to make software, which is originally designed for English speaking people, also usable for those whose primary (or only) language is not English as described in detail in Ref. 7. IBM Corporate Standards are in place for many years by now, which require IBM software developers to include DBCS handling and other relevant considerations from initial design so that their products can be accepted in the Far Eastern countries

Multilingual Applications in Future PCs

Multilingual text processing on different PCs led me to conclude that using a Japanese version PC was advantageous to write this book. It can handle both, single and double-byte and with unlimited use of, for example, all features of MS Word and Excel. It would be ideal, therefore,

to eventually equip all PCs in the future with double-byte, multilingual capability, based on a uniform architecture.

Such a future PC must allow the users to toggle between language modes for tool bars, pull-down menus, messages, and help information. Users should be able to choose the languages they wish to toggle through. It would require loading dictionaries of several major languages, but hard disk storage space becoming less expensive and more abundant should allow this.

Since my retirement from IBM in 1992, I have been working with many PCs, and mostly IBM products. I also often work with files in three different languages - English, German, and Japanese, which causes a lot of frustration. In many applications, two, or all three languages appear in the same file, and I am sending many of them via the Internet.

This book, being bi-lingual, was produced on T21 and later G40 IBM ThinkPads, Japanese models. Producing text in English is simple with the Japanese version of Microsoft Word and Excel. However, all software tools are in, such as pull-down menus, error messages, and help functions. Thus, a user must be fluent in reading and writing Japanese, which cannot be expected in many cases. It would be most welcome, if the language mode for these functions could be made selectable from a list of (major) languages.

Code Page Structure Control

German "Umlauts," or in general European accented characters, are difficult to enter, with methods being different by software maker. I am using a so-called language kit on my US-based PC to enter them with control-key combinations. This, however, does not work on my Japanese PC that runs the Japanese version of MS Word. It can handle English, but not the German accents. Five click levels are needed to reach the symbol table to copy them from there. But, when I finally paste them into my file, some Kanji characters appear instead.

Code page conflicts apparently exist between single and double-byte PCs when transferring files via Internet. When I send a file from my US PC with German text (upper box in Figure 53) to someone receiving it via Eudora on a Japanese PC, the German "Umlauts" (blue) appear as Kanji or special characters, lower box (red).

There is a definite need for compatibility on an international scale. Diverging approaches by different software makers should be eliminated and replaced by a standard approach. So-called "handshaking" requirements between sender and receiver to ensure proper encoding and decoding must not be imposed on the user. They should be detected by the affected software and be adjusted automatically.

Before Unicode was invented, there were hundreds of different encoding systems, many conflicting with one another. Any given computer (especially servers) needs to support many different encodings; yet, whenever data are passed between different encodings or platforms, there is a risk that data may be corrupted. Although with Unicode as the official standard to implement ISO 10646 now in place, millions of computers are still running back-level software that does not support Unicode. My example is such a case. It will take many years until the world is using only fully Unicode supporting computers.

Es sind aus allen Ländern Firmen da die den Ablauf bestimmen und prüfen. Englisch ist hier die 2.te Sprache in Wort und Schrift. Wir "Langnasen" wie sie uns hier nennen wurden ständig von den jungen Leuten angehalten um viel von unserem Land zu erfahren. Die Leute sind nett und sehr freundlich. Es gibt an jeder Ecke offene Kleinküchen mit allem Möglichen zu essen. Die jungen Leute sind sehr modisch gekleidet. Im Fernsehen werden 15 Programme per Satellit gesendet. China hat sich geöffnet. Wer eine Einladung und Geld hat kann Reisen, wohin er will. Unsere Reise ging nun von Berlin/München-Shanghai. 3Tage.Wuhan-5Tage Yichang-Kreuzfahrt Xi` 工 an-Peking. Nächster Bericht folgt. Grüße an Alle.
Es sind aus allen L 舅 dern Firmen da die den Ablauf bestimmen und pr 鬱 en. Englisch ist hier die 2.te Sprache in Wort und Schrift. Wir "Langnasen" wie sie uns hier nennen wurden st 舅 dig von den jungen Leuten angehalten um viel von unserem Land zu erfahren. Die Leute sind nett und sehr freundlich. Es gibt an jeder Ecke offene Kleink ・ hen mit allem M ・ lichen zu essen. Die jungen Leute sind sehr modisch gekleidet. Im Fernsehen werden 15 Programme per Satellit gesendet. China hat sich ge ・ fnet. Wer eine Einladung und Geld hat kann Reisen, wohin er will. Unsere Reise ging nun von Berlin/M ・ chen-Shanghai. 3Tage.Wuhan-5Tage Yichang-Kreuzfahrt Xi` 工 an-Peking. N 臘 hster Bericht folgt. Gr ・ e an Alle.

Figure 53. Coding Corruptions via E-Mail Source: Ref. 20

Multilingual Operating Systems

Providing multi-lingual operating systems with user-selectable languages for the tool bar, pull-down menus, prompts, error messages, help functions etc. would be extremely helpful in cases like writing this book in different countries on different computers. I am afraid it will

take several more years until this would be available, but also affordable for all users to replace back-level systems.

E-Mail Processing

Producing this book required the use of several computers, not always my own, while in the US, Japan, and Germany. In order to always have access to all related correspondence (with attached files), I needed to take along a CD with that information. The ensuing problems go beyond Kanji, but are worth mentioning.

I found myself wasting precious time switching around between Eudora, Netscape, Outlook Express, CompuServe, AOL or others. The message file folder organization, structures, address books, identity management data, protocol routines, bookmarks and other parameters are all different and very hard to find when attempting a transfer from one system to another. I am sure there are other Internet users who would have nightmares if they wish to do this. There is a definite need for making interaction between PC users and providers simpler, more user-friendly, and most importantly, uniform. A set of international standards would certainly help.

Acknowledgements

Karl Ganzhorn

Karl Ganzhorn is first for me to mention on this list, because I consider him my mentor in all aspects. His leadership and support during the early years of my association with IBM moved me up in my career and my entire life. Although we were separated later by location and reporting line, our contacts remained alive. While visiting him at his home during a trip to Europe during the summer of the year 2000, he twisted my arm to write a book on the topic "IBM History of Far Eastern Languages in Computing." I said "yes."

Rolf Gaebele

My friendship with Rolf Gaebele developed over decades of a business relationship. I was following his activities in several national and international committees that established standards in the information technology area. After retiring from IBM Germany, Rolf became active in adult education programs in Germany. Rolf provided me with very helpful information to clarify the complex fuzziness of multilingual Internet applications.

Toshiaki Igi

Toshiaki Igi joined Masumi Iwao's small IBM Japan Engineering Liaison group in 1966, and which he took over later. He spearheaded the efforts to solve the Far Eastern Language requirements in IBM products and was the first manager of the Double-byte-character-set Technical Coordination Office (DTCO) when it was established in 1982. Now retired from a high level position at Yamato Lab, he contributed invaluable material from his archives, carefully kept over decades of his leading role in the development of Far Eastern Language capabilities in IBM products. This book as it is would not have been possible without Toshiaki's help.

During one of my visits to Japan in November 2000, he organized a special meeting for me at the IBM Japan Yamato Product Development Laboratory. Several young men attended, most of them from the "Globalization Center of Competency - Yamato", formerly his DBCS Technical Coordination Office. Some retired old time associates who were involved in this effort came too. I felt like working for IBM again, ten years after retirement, and left with a wealth of information, useful to trace back and document forty years of IBM progress in Far Eastern languages computing. Since that meeting, Toshiaki worked very closely

with me to produce this book. He provided the detailed input about IBM activities in which he had participated. He played the pivotal role in organizing the right people in IBM for finding answers to my questions. My thanks go to Toshiaki Igi for his valuable contributions, with a request to forward my gratitude to the many individuals who provided the help that I needed for turning this book into an appreciable piece of documentation.

Masumi Iwao

Holding an Engineering degree in Precision Mechanics, Masumi Iwao worked with Canon Company a few years before joining IBM Japan in 1961. He developed the Katakana Feature for the IBM 1440 Printer and became the lead man of the small Engineering Liaison group in 1964, assisting me to address the language problem. "Massi" also played a major role in the development of the first IBM Kanji Data Processing System. Four decades later, he enthusiastically gave his support for writing this paper and lined up people whose contributions were needed to make this effort worthwhile. Massi's assistance was irreplaceable for the translation of relevant information that was available only in Japanese. Upon completion of the book, Massi arranged for a nostalgic reunion in Tokyo, to which we invited Toby Takahashi (the first Japanese whom I met arriving in Japan 40 years ago). The picture shows (from left to right) Toby, Toshiaki, Kurt, and Massi.

Akira Oda

Akira Oda holds a Masters degree in Electric Engineering from Kyoto University. He and his team, Globalization Center of Competency, Yamato Software Lab, IBM Japan, made outstanding contributions on DBCS standardization, and he provided the summary information on the development of all relevant documents on coded character sets.

Toru Takeshita

Toru Takeshita is a former colleague but also a long-time friend. He graduated from the mathematics department of Kyoto University, joined IBM Japan in 1957, and participated in the development of the first IBM COBOL compiler from 1960 to 1962. He led a programming team to develop the IBM Tokyo Olympic Information System from 1962 to 1964, and worked on software requirement, development, and marketing and research areas. After retirement from IBM in 1991, he was a professor of Information Science at Chubu University, Nagoya, Japan, until March 2004. He is a member of the IEEE Computer Society, and honorary member of IPSJ (Information Processing Society of Japan).

Toru said my book would become a highly valuable historical document in computing for academia and industry, which encouraged me to pursue the project. He provided me with corrections, additions and leads to other sources of information. I worked them all into this book.

Akio Kido

Akio Kido spent considerable time reviewing and commenting on my earlier draft via e-mail. He also referred me to relevant recent publications and key people within IBM Corporation as well as international standards committees. His inputs contributed considerably to the substance and accuracy of this book.

Satoru Kimura

Satoru is a brilliant, young, staff engineer with Product Assurance, PC Development, Asia Pacific Technical Operations (APTO), IBM Japan Yamato Lab. Assisted by Ken Han, on assignment from IBM China, Satoru was of tremendous help to me not only to understand the complexities of Traditional and Simplified Chinese, but also the data entry implementation. Most of the figures in the Chinese Data Entry section of my book were skilfully captured on the fly by Satoru while meeting with him in his office.

Ranat Thopunya

Ranat is the IBM Thailand, Manager of the National Language Development & Translation Services Center (GCoC-CTL, or Globalization Center of Competency - Complex Text Languages). With first-class responsiveness, he assisted in compiling and clarifying the approach that IBM Thailand undertook to obtain proper language implementation.

Ken Lunde

Ken Lunde of Adobe Systems is the author of *CJKV Information Processing* (1999). I saw its description on the Internet and decided immediately to buy his book. It provides a wealth of information and analyses of the major Far Eastern languages that I would have loved to have forty years ago. I used it daily to refresh my memory while writing this book, in order to verify the accuracy of my statements. Ken Lunde graciously granted me his permission to include some of his information in this book.

CP Chang

CP Chang is Manager of Globalization, IBM China, Shanghai. He deserves many thanks for his efforts to provide me with archive information on early implementation of Chinese on IBM products and the present IBM ThinkPad.

Dongchang (D.C.) Lee

D.C. Lee is the Manager, IBM Korea Translation Services Center (TSC). He was very helpful in providing me with information on Korean data entry, and for researching archive information of early Korean language implementation.

Takao Hensch

My son gave me immense background support in the form of hardware, software, formatting advice, printing, and airline mileage. Without his logistical help, it would have been next to impossible to come up with this book in the shape it is, and under the time-frame as I did. Takao is an example for an ideal, close and happy father – son relationship

Chigusa Ishikawa Kita

Chigusa holds a PhD on "The History of Computer Development Concepts in the US" from the Kyoto University Post Graduate School of Literature. The IEEE had selected her, a Japanese native, to be one of the first reviewers of this paper. Her comments made it possible to properly describe IBM's role in context with the efforts of national companies.

List of Figures

Continued on next Page →

List of Figures (continued)

Kurt Hensch

With an Engineering degree in Precision Mechanics, Kurt Hensch joined the IBM Germany Development Lab in 1957. An assignment to IBM World Trade Corporation HQ, New York in 1961 was the beginning of his international career; five years at IBM Asia Pacific HQ, followed by IBM World Trade, and Corporate HQ until his retirement as Program Director of Standards in 1992.

kurt@kdd.net

References

1. IBM Japan 50th Anniversary Chronicles in Publication "ACCESS", April 1981 edition (in Japanese), Fig 5-9 on page 325.

2. K. Lunde: CJKV Information Processing, Chinese, Japanese, Korean, and Vietnamese Computing, O'Reilly & Associates, 1999

3. K. Fowels: "Unicode Evolves", BYTE, March 1997, pp. 105-110

4. The Unicode Standard, Version 3.0, The Unicode Consortium, Addison-Wesley Longman, Inc., 2nd Printing April 2000

5. K. Shibano (Ed): JIS Kanji Dictionary, JSA (in Japanese), 1997

6. Y. Mikami: The History of Character Codes -- Asia (in Japanese), ISBN4-320-12040-X, Kyouritsu Shuppan, 2002

7. Y. Kiyokane and Y. Suehiro (Eds): Internationalization Programming -- I18N Handbook (in Japanese), Kyouritsu Shuppan,1998

8. JIS X 0201:1997 7-bit and 8-bit Information-Interchange Coded Character Set (in Japanese), JSA, 1997

9. JIS X 0208:1997 7-bit and 8-bit Information-Interchange Coded Kanji Set (in Japanese), JSA, 1997

10. JIS X 0212:1999 Information-Interchange Kanji Code -- Additional Kanji (in Japanese), JSA, 1999

11. JIS X 0213:2000 7-bit and 8-bit Information-Interchange Coded Expanded Kanji Set (in Japanese), JSA, 2000

12. S. Zhai, "What's in the Eyes for Attentive Input", Communications of the ACM , March 2003/Vol. 46, No. 3, pages 34-39

13. T. Igi, documents archives

14. Brochure "IBM Kanji Information Processing System" published by IBM Japan Ltd. 8-71, N:G518-5001-0

15. Manual "IBM Japanese Information Processing System/General Description", published by IBM Japan Ltd. 8-71, N:GA18-1160-0

16. "Logischer Entwurf eines digitalen Rechengeräts mit mehreren asynchron laufenden Trommeln und automatischem Schnell-speicherbetrieb", Dissertation Technische Universität Berlin, 1957

17. IBM Thailand Globalization Center of Competency

18. IBM Japan Globalization Center of Competency

19. IBM Japan Product Assurance

20. K. Hensch, documents archives

21. Japan TV commercial in USA

22. Kanji Character Set and Code Reference Manual for IBM 2245 Kanji Printer System, published by IBM Japan Ltd. 9-73, (N:GA18-1018)

23. Kanji Character Set and Code Reference Manual for IBM 5550 Kanji Work Station, published by IBM Japan Ltd. 1973, (N:GC18-2040)

24. Kanji Character Set and Code Reference Manual for IBM Kanji Systems, published by IBM Japan Ltd. August 1979, (N:GC18-0611)

25. Kanji Character Set and Code Reference Manual for IBM Kanji Systems, published by IBM Japan Ltd. 1986, (N:GC18-0785)

26. Toru Takeshita: "Programming Lab for Junior High Students" (in Japanese), Suugaku (Mathematics) Seminar, Vol. 8, No.11, Nihon Hyouron Sha (Japan Review Co.), Nov. 1969

27. IBM System/370 – DBCS Application Primer (Enabling your programs for Chinese/Japanese/Korean), GC18-9059, published by IBM Japan Ltd. Nov 1987

28. DBCS Design Guide – IBM System/370 Software, C18-9095, published by IBM Japan Ltd. June 1988

ISBN 3-937267-03-4